Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1988

An expert system to detect espionage through credit record analysis

Salazar, Gary M.F.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/23039

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

S15265

AN EXPERT SYSTEM TO DETECT ESPIONAGE
THROUGH CREDIT RECORD ANALYSIS

by

Gary M.F. Salazar

March 1988

Thesis Advisor: Taracad R. Sivasankaran

Approved for public release; distribution is unlimited

T239271

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited.
2b DECLASSIFICATION DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (*If applicable*) Code 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, California 93943-5000		7b. ADDRESS (*City, State, and ZIP Code*) Monterey, California 93943-5000
8a NAME OF FUNDING SPONSORING ORGANIZATION	8b OFFICE SYMBOL (*If applicable*)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)		10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11 TITLE (*Include Security Classification*)

AN EXPERT SYSTEM TO DETECT ESPIONAGE THROUGH CREDIT RECORD ANALYSIS.

12 PERSONAL AUTHOR(S) Salazar, Gary M.F.

13a. TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM ______ TO ______	14. DATE OF REPORT (*Year, Month, Day*) 1988 March	15 PAGE COUNT 156

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
Expert system; Credit; Finance; Credit Analysis; Risk; TRW; Espionage

19 ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Finance has been the prime motivation in many recent espionage cases. This thesis expanded the prototype Manpower Financial Tracking Expert System (MFTES) that analyzes individual financial profiles in order to detect the potential of an employee to engage in espionage activities. The architecture of MFTES has modules that capture Control Strategy, Taxonomy of Concepts Expert Rules, and Numerical Processing. During evaluation, it successfully followed the documentary trail for 75 employee credit reportsand made inferences about their potential risks.

20 DISTRIBUTION AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Taracad Sivasankaran	22b. TELEPHONE (*Include Area Code*) (408) 646-2637	22c. OFFICE SYMBOL Code 54Si

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted.
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

☆ U.S. Government Printing Office: 1986—606-24.

Approved for public release; distribution is unlimited.

An Expert System to Detect Espionage Through Credit Record Analysis

Gary M.F. Salazar
Lieutenant Commander, United States Navy
B.S., University of Colorado, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1988

ABSTRACT

Finance has been the prime motivation in many recent espionage cases. This thesis expanded the prototype Manpower Financial Tracking Expert System (MFTES) that analyzes individual financial profiles in order to detect the potential of an employee to engage in espionage activities. The architecture of MFTES has modules that capture Control Strategy, Taxonomy of Concepts, Expert Rules, and Numerical Processing. During evaluation, it successfully followed the documentary trail for 75 employee credit reports and made inferences about their potential risks.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND 4

- A. HISTORY OF EXPERT SYSTEMS 4
- B. SCOPE 7

III. IMPLEMENTATION 11

- A. KNOWLEDGE REPRESENTATION OF THE FINANCIAL PROFILE 11
- B. ARCHITECTURE OF MFTES AND EXAMPLE CASES . . . 14
- C. RULE FORMULATION AND STRUCTURE 17
- D. MFTES OPERATION 19
- E. CHARACTERISTICS OF FINANCIAL PROFILE ANALYSIS. 20
- F. ANALYSIS FOR BANKRUPTCY 22
- G. ANALYSIS FOR SUPERPAYER 25
- H. ANALYSIS OF OVERALL WEIGHT VALUE 29

IV. EVALUATIONS, CONCLUSIONS AND RECOMMENDATIONS . . . 33

- A. EVALUATION RESULTS 33
- B. HARDWARE AND SOFTWARE REQUIREMENTS 34
- C. SUCCESSES 35
- D. LIMITATIONS AND FUTURE IMPROVEMENTS 36

APPENDIX A: SOURCE CODE 39

APPENDIX B: TRW CREDIT REMARKS 143

LIST OF REFERENCES 147

INITIAL DISTRIBUTION LIST 149

LIST OF FIGURES

Figure 3.1 MFTES concept relationships 13

Figure 3.2 Architecture of MFTES 14

Figure 3.3 Sample interaction with MFTES 16

Figure 3.4 Rule base structure for credit remarks . . 18

Figure 3.5 MFTES case format 21

Figure 3.6 Flowchart for Bankruptcy Analysis 24

Figure 3.7 Monthly Discretionary Income 26

Figure 3.8 Sample case with weights and cutoff levels 32

I. INTRODUCTION

The use of espionage by hostile forces to gain access to classified information is a clear threat to our government. The effect of espionage is not always readily apparent but there can be significant and irreparable damage to our national security. The motives for espionage are many. Whether it is committed for money, personal gain or for political reasons, the prevention of espionage has become a high priority of our security forces. In most cases, including those of convicted spies FBI agent Richard Miller, the National Security Agency's Ronald W. Pelton, and the Navy's John Walker, financial gain has been the prime motivation. [Ref. 1]

Enormous resources are required to detect and prevent espionage. The money spent and manpower employed to identify, locate and then apprehend a spy can only be justified when compared to the loss of information vital to our national security. Miller, Pelton and Walker, and other similar cases, created clearly identifiable documentary trails. Rosa concluded that if there had been a computer system automatically tracking individuals' income and spending, it could have alerted the authorities in time to limit, if not prevent, the resulting damages [Ref. 2].

Currently the Defense Investigative Service (DIS) obtains paper credit reports on subjects undergoing background investigations for positions requiring top secret clearance. These documents are reviewed for derogatory information revealing financial irresponsibility, which may be cause for denial or revocation of clearance eligibility by the cognizant adjudicating authority. Under current procedures all credit reports, both those with varying degrees of negative information and those that are 'clean', are produced in paper format and individually reviewed by DIS personnel. It will become increasingly important for DoD to automate this process as much as possible since stabilized or even decreasing human resources are now anticipated. [Ref. 3]

At the same time the role of finances in so many recent espionage cases has led to calls for increased use of credit checks, even to the point of including them as part of investigations for secret level clearances. A 1985 report to the Secretary of Defense by the commission to review DoD security policies and practices recommended expansion of the investigative scope for a SECRET clearance to include a credit check of the subject [Ref. 4, p. 9].

Clearly desirable is a streamlined process in which the generation of paper is kept to a minimum and human resources are devoted only to those cases with derogatory indications [Ref. 3].

The purpose of this thesis has been to more fully develop the prototype MANPOWER FINANCIAL TRACKING EXPERT SYSTEM (MFTES). MFTES analyzes individual financial profiles in order to detect the potential of an employee involving himself in espionage activities. The prototype was developed by Sivasankaran and Bui [Ref. 1] of the Naval Postgraduate School (NPS), Monterey, California. The prototype was sponsored by the Defense Personnel Security and Research Center (PERSEREC) as part of its effort to improve the DoD personnel security program through increasing the use of automated data and techniques.

The thesis is organized as follows: Chapter II provides background on the evolution of expert systems, on the MFTES prototype and its expanded development. The MFTES implementation is discussed in Chapter III, followed by a discussion of the evaluation tests, the expanded system's hardware and software requirements and the conclusions and recommendations in Chapter IV. Appendix A is a listing of the MFTES source code and Appendix B is a listing of the 102 account conditions (remarks) used in the TRW credit rating system.

II. BACKGROUND

A. HISTORY OF EXPERT SYSTEMS

"An expert system is a method for handling real-world, complex problems requiring an expert's interpretation and which solves these problems using a computer model of expert human reasoning. It reaches the same conclusions that a human expert would reach if faced with a similar problem." [Ref. 5] The heuristics and rules that make up a human expert's knowledge are transferred into the rule-base and inference engine of the expert system. The facts within the rule base can be represented in many ways but are usually in the form of an IF...THEN relationship. The inference engine contains the strategies by which the expert system orders the rules, facts and goals to reach a conclusion.

Expert systems have been in use for over 15 years and have roots to the pre-World War II period with the development of Formal logic and Cognitive psychology. Since that beginning, expert systems have been used in many different types of commercial applications. Two early, successful systems developed at Stanford University were DENDRAL and MYCIN. DENDRAL, conceived in the 1960's, is a chemistry expert system designed to examine a spectroscopic analysis of an unknown molecule and predict the molecular structures that could account for that particular analysis. MYCIN, developed in the mid-1970's was designed to aid physicians in the diagnosis and treatment of meningitis and bacteremia infections. [Ref. 6, p. 15] Commercial applications in the 1980's cover many different fields including decision making, software maintenance management, software design and development aids, ocean surveillance, knowledge-based tutors, information management and weather forecasting. Applications in the military are just as widespread. Some of the more recent efforts include RICA: an expert system for radar image classification, ACES: an airborne communications expert system, and TARSIA: a system that helps track underwater contacts. [Refs. 7 and 8]

Practical applications for expert systems abound. Whenever human experts are in great demand and short supply, a computer based consultant can help amplify and disseminate the needed expertise. An expert system can capture the practical experiential knowledge that is hard to pin down. [Ref. 9]

Credit institutions have always used heuristic procedures in their analysis of individual financial profiles in order to assign good or bad risk ratings. Although such analysis is not considered exact, enough advances in the field have been made to make it practical to articulate the heuristics in the examination and interpretation of financial data. Many of these heuristics appear structurable enough to be captured by the current expert system technology. [Ref. 1]

One industry, financial services, is emerging as the next significant user of expert systems. One financial services leader in expert systems implementation is New York-based American Express Co.
The company began its involvement with its current AI expert system approximately 2 1/2 years ago. It uses a customized expert system called the Authorizer's Assistant (AA), which was designed to help the authorizer filter through credit data. American Express has noted three primary benefits derived from the AA system. First, productivity and time savings. The company expects 20% time reduction in credit review and has observed a 96.5% accuracy rate to date. Second, losses associated with bad credit risks have been reduced by the improved screening process. Third, improved customer service benefits are expected because the higher percentage of quicker approvals will most likely increase customer satisfaction. [Ref. 10]

Another expert system user, Equitable Financial Cos., has produced a demonstration model that evaluates the underwriting risk of someone who might have a history of alcohol abuse. Using information obtained from employers, doctors and so on, the system helps identify high-risk individuals by reading between the lines. The R&D model is generic enough that, with different rules, it can be applied to other underwriting problems. [Ref. 10]

B. SCOPE

One of the responsibilities of PERSEREC is to investigate the feasibility of using existing financial data bases to monitor the financial health and behavior of individuals holding security clearances. [Ref. 11, p. 3] Information containing credit histories has already been implemented and distributed nation wide through large database systems such as the ones managed by TRW [Ref. 1]. "While one of about half a dozen credit report vendors currently used by DIS, TRW was selected for this study because of its national coverage and existing GSA contract...." [Ref. 3].

TRW uses 102 credit remarks which can be attributed to an individual's credit standing. All were used in the MFIES expansion and are included in Appendix B. Examples of some of the major remarks that are reported by credit bureaus are listed in Table I.

TABLE I. TRW CREDIT REMARKS USED IN MFTES

Report Abbreviation	Explanation
INQUIRY	A copy of the credit profile has been sent to this credit grantor at their request.
BK LIQ REO	Debt included in or discharged through Bankruptcy Chapter 7 or 11.
BK 7 FILE	Voluntary or involuntary Petition in Bankruptcy; Chapter 7-(Liquidation) filed.
BK 7 DISC	Voluntary or involuntary Petition in Bankruptcy; Chapter 7-(Liquidation) discharged.
CO TAX LN	County Tax Lien.
PD COLL AC	Paid account/was a collection account insurance claim or education claim.
PD WAS 180	Paid account/was delinquent 180 days.
FORECLOSURE	Credit grantor sold collateral to settle defaulted mortgage.
REPO	Merchandise was taken back by credit grantor; there may be a balance due.
CHARGE OFF	Unpaid balance reported as a loss by credit grantor.
JUDGMENT	Judgment.
COLL ACCT	Account seriously past due/account assigned to attorney collection agency or credit grantor's internal collection department.
DELINQ 180	Account delinquent 180 days.
DELINQ 120	Account delinquent 120 days.
DELINQ 60	Account delinquent 60 days.
30 DAY DEL	Account past due 30 days.

PROLOG was chosen as the programming language because it is a language currently used in a wide spectrum of expert system applications. [Ref. 12] Additionally, PROLOG is integrated within the Arity/Expert Development Package, a commercially produced collection of development tools which provide the basis for constructing an expert system [Ref. 12] The Arity/Expert Development Package was chosen as a shell for MFTES because of its ability to handle medium to large size expert systems, its control mechanism for controlling the way in which information is ordered or accessed, and its facility for presenting explanations to the reasons behind decisions. "Arity is one of the leading companies in the Prolog world with a strong reputation for reliability and support...and the package has a proven record of commercial success" [Ref. 13].

Initial consideration was given to developing the expanded MFTES on the NPS mainframe computer because an interface to TRW Credit System reports, also generated on a mainframe, could then be more easily achieved. However this proved infeasible because the Arity/Expert Development Package is not configured to function in a mainframe environment. A local alternative was IBM's own expert system development application, known as Expert System Environment (ESE), which was produced specifically for the mainframe environment. The ESE package was temporarily on loan to the Naval Postgraduate School for a six month trial period. ESE was considered as a candidate for the MFTES expansion but its potential loss after six months and the lack of local technical support suggested the more prudent decision was to continue using the Arity/Expert Development Package. This also meant the development effort could continue from the point where the MFTES prototype left off rather than having to duplicate code already produced.

Another reason for the choice of a microcomputer configuration was the eventual environment in which MFTES would be used. MFTES was intended for use by government security officers or employees responsible for reviewing personnel credit backgrounds, and because they could be expected to operate in a variety of different locations (buildings, ships, etc.), the hardware configuration was expected to portable, readily available and easy to use. The personal computer (PC) appeared perfectly suited for those conditions.

The only funding used was for record typing services which converted 75 actual TRW records to a format usable by MFTES, and for financial consultant services that helped define the heuristics applied by MFTES.

III. IMPLEMENTATION

A. KNOWLEDGE REPRESENTATION OF THE FINANCIAL PROFILE

MFTES represents financial analysis expertise through concepts captured by frames. Concepts are defined by their qualities and the possible values or properties of those qualities. For instance, the concept Superpayer may have properties such as monthly payments, that range up to $20,000, and the number of active creditors, which goes up to 100. Figure 3.1 shows how the knowledge and relationships of different concepts are represented in the system.

Concept relationships are represented using the notion of a role. For example, the concept PERSEREC is related to the concepts CUSTOMER_PROFILE, SUPERPAYER and BANKRUPT through the roles CUSTOMER_PROFILE_INFO, SUPERPAYER_INFO and BANKRUPT_INFO. The concept PERSEREC holds information on the profile of the customer, the possibility of his being a Superpayer and his potential for bankruptcy. Concepts may be connected through a series of intermediary concepts.

Since the connections are captured through the use of roles, a 'role-chain' then strings the various concepts in a hierarchical fashion. The lowest level concept at the end of the chain is represented in terms of its properties or values.

It is also possible for a lowest level concept to have more than one specific value. For instance, the concept CONCLUSION can be represented with one of the following values: TOO MANY ACCTS, SUPERPAYER, NORMAL, SATISFACTORY, POOR, VERY POOR, SERIOUS, CRITICAL, VERY CRITICAL, and POTENTIAL BANKRUPT. [Ref. 1, pp. 14-18]

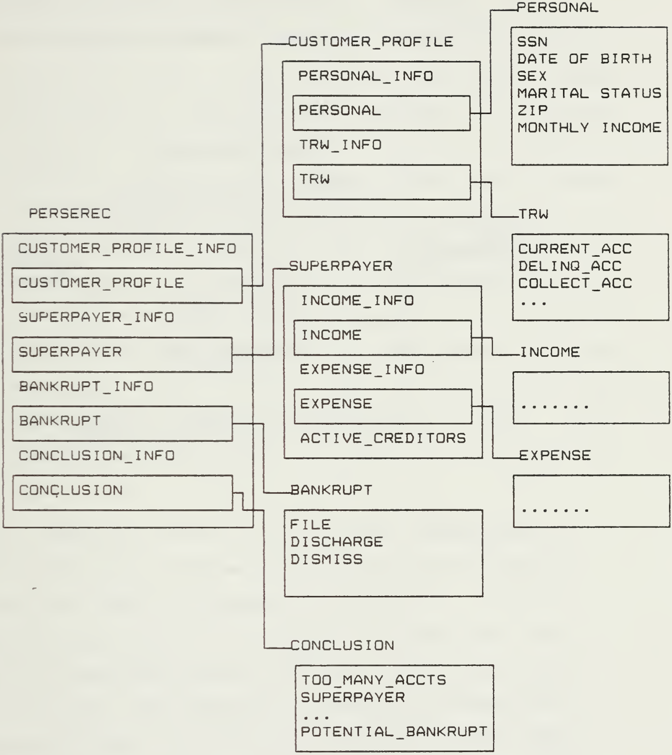

Figure 3.1 MFTES concept relationships

B. ARCHITECTURE OF MFTES AND EXAMPLE CASES

The functional architecture of MFTES is illustrated below in Figure 3.2. The expert system has five modules.

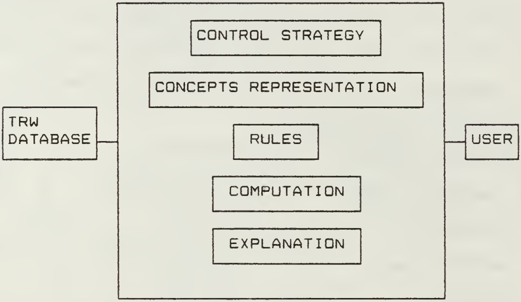

Figure 3.2 Architecture of MFTES

The first module consists of the Control Strategy to be employed in manipulating the rules during a problem solving situation. The Control Strategy also determines when to access the static knowledge in the Concepts Representation module and when to access the dynamic knowledge contained in the Computation module. The system takes one individual credit report at a time as its input from the TRW database and provides its conclusion to the user. An Explanation module is also available to provide details of how a conclusion was reached.

A sample interaction with the system is shown in Figure 3.3 [Ref. 1, pp. 14-18]. It shows a NORMAL conclusion was reached because the weight of the current case was found to be EXTREMELY LOW. The determination of EXTREMELY LOW was made because the case's OVERALL CALCULATED WEIGHT was found to be 10 which was less than the CUT OFF VALUE of 40. The OVERALL CALCULATED WEIGHT was determined to be 10 because the PRIVATE RECORD of the case contained that value. A closer examination of the PRIVATE RECORD showed that there was some MISCELLANEOUS ITEM which was assigning the 10 points. Deeper interaction with the system would determine precisely what credit remark contained in this record was causing the value of 10 to appear.

```
  The person is :
                               1) normal / 1.0
  [ How was  the conclusion  of  the current case -2
  judged to be normal (1.000) ? ]

  Since [1]  the above weight of  the current case on
            a range-2 is extremely low (1.0)
            then the conclusion  of the current case
           -2 is normal (1.0)


         previous
         explain
         continue
  [ How was above weight of the above case on a
  range-2 judged to be extremely low (1.000) ? ]
  Since [1]  the overall calculated weight of the
        current case -2 is 10 (1.0) and
        [2]  the extremely low cut off value of the
          range 0-200 that we use-1 is 40 (1.0)
                  and  10 =< 40
          then the above weight of the current
             case on a range-2 is extremely low (1.0)
  [ How was overall calculated weight of the current
case -2 judged to be 10 (1.000) ? ]

  Since [1]  the public record val of the current
             case -2 is 0 (1.0) and
        [2]  the private record value of the current
           case -2 is 10 (1.0)
                  and  10 = 0 + 10
           then the overall calculated weight of the
           current case -2 is 10 (1.0)
  [ How was private record value of the current case
 -2 judged to be 10 (1.000) ? ]
  Since [1]  the calculated weights for all current
          and paid accounts of the current case
             -2 is 0 (1.0) and
        [2]  the calculated weights for all
             delinquent accounts of the current case
           -2 is 0 (1.0) and
        [3]  the calculated weights for all
             miscellaneous private items of the
      current case -2 is 10 (1.0) and
             10 = 0 + 0 + 10
             then the private record value of the
        current case -2 is 10 (1.0)
```

Figure 3.3 Sample interaction with MFTES

C. RULE FORMULATION AND STRUCTURE

The problem-solving heuristics involved in the financial analysis are captured in the form of rules. Rules help the expert system manipulate the concepts, their properties and values so a conclusion about the individual credit report can be developed. Rules were represented using consequents and antecedents. A consequent consists of a single goal, whereas the antecedent can be made up of one or more goals. A goal indicates a property of a concept has some particular value. [Ref. 1, pp. 14-18]

In MFTES a simple weighting scheme was interleaved with many of the rules to add up a preset number of points for each negative remark occurring in the credit record. The idea is similar to the scheme employed in traffic violations. An overall conclusion is based on total points scored on the credit record. [Ref. 1, pp. 14-18]

Rules were written following the same structure shown in TRW's Glossary of National Status Comments [Ref. 14]. This explanatory aid for deciphering the coded report abbreviations, divides the 102 credit remarks used by TRW for its reports into two broad categories, Items of Private Record and Items of Public Record. Figure 3.4 shows the MFTES rule base structure further subdividing the Private and Public Records into smaller sets containing similar remarks.

The Public Record is made up of a two sets, one addressing bankruptcy and liens and another dealing with related miscellaneous items. Remarks grouped within the bankruptcy set included bankruptcy adjustment plans, bankruptcy chapters 7, 11 or 13, mechanic's liens and federal or city tax liens. Remarks within the miscellaneous items set included judgments and suits.

The Private Record is made up of three sets, one addressing current and paid accounts, one dealing with delinquent accounts and the other also dealing with miscellaneous items. Remarks within the current and paid accounts set included all paid accounts past due and current accounts that were collections. Remarks within the delinquent accounts set included delinquencies 60 days or greater and foreclosures. Remarks within the miscellaneous items set included settled accounts and charge offs now being paid.

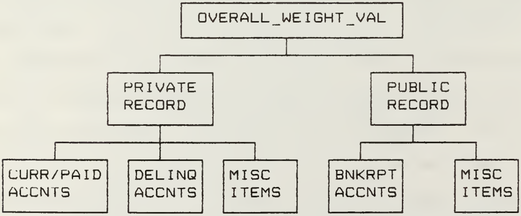

Figure 3.4 Rule base structure for credit remarks

D. MFTES OPERATION

MFTES examines a TRW report by first determining if an individual's credit profile is one that indicates he is already bankrupt or in the process of bankruptcy. Next it determines if potential bankruptcy is indicated or if a superpayer condition exists.

Anyone who is bankrupt or going through the bankruptcy process is considered a high risk. The loss of financial well being combined with access to classified information makes any such individual vulnerable to compromise. The sale of classified information for money, in order to improve financial position, is a circumstance that might be avoided if the bankruptcy condition is identified early enough.

The same reasoning applies to potential bankruptcy. Anyone finding himself going so deeply into debt that bankruptcy could become a reality is also considered a risk for potential espionage.

A superpayer is someone sitting at the opposite extreme from bankruptcy. His financial position appears so good that bankruptcy is never a question. A superpayer's characteristics, however, should raise anyone's eyebrows. His credit information might appear flawless showing timely payments, paid off debts and a good rating. But a comparison of a superpayer's monthly expenses against his expected monthly income would reveal that his expenses exceed his income. The rationale is that if someone is spending more than it appears he could afford, there is an external source of income which the individual is enjoying that needs close examination. [Ref. 1, p. 8] The superpayer is also considered a high risk.

If any of the bankruptcy, potential bankruptcy or superpayer conditions are not initially identified the MFTES then evaluates the individual's credit record for his overall financial position. A weighting scheme assigns point values (weights) to each of the different credit remarks found in the credit report. These weights are used to indicate the individual's financial position on a predefined scale that ranges from Normal to Very Critical.

E. CHARACTERISTICS OF FINANCIAL PROFILE ANALYSIS

The objective of financial profile analysis was to ascertain what financial remarks were present in a credit report and then, based on the number and combination of those remarks, make inferences about the individual's overall financial position. [Ref. 1, p. 6] The financial profile analysis was oriented towards discovering two kinds of signals, whether an individual was stretching beyond his financial means and was a candidate for bankruptcy, or whether the individual was spending a disproportional amount of his income. [Ref. 1, p. 6]

It should be noted that TRW credit records are stored on magnetic tape and these had to be translated into a format usable for MFTES in a microcomputer environment. Applicable portions of each TRW credit record examined were extracted and reformatted into a format easily utilized by the MFTES. Figure 3.5 shows the employee and data headings of an actual case (The SSN has been altered) in that format.

```
EMPLOYEE(SSN,GSLEVEL,GSSTEP,MARITAL STATUS,ZIP CODE)

        TYPE DATE  DATE TYPE           AMOUNT
 DATA( ,RMRK,RPRTD,OPEN,ACCT, ,AMT,BAL,PAST DUE,_)

 employee('000-00-0000',9, ,married,'93943').
 data(_,curr_acct,1085,682,i,985,3845,924,0,0).
 data(_,curr_acct,1185,380,r,_,780,400,25,0).
 data(_,'30_day_del',985,1179,r,_,1759,1682,157,1).
 data(_,curr_acct,985,1283,r,685,584,584,54,1).
 data(_,curr_acct,585,1176,r,485,2861,0,0,0).
 data(_,curr_acct,885,1184,r,585,700,807,15,0).
```

Figure 3.5 MFTES case format

Some financial information, such as an employee's grade and step data, is not contained on the TRW credit report and was added to the new format for use in determining monthly income. This specific type of salary information is available from a personnel record or database.

F. ANALYSIS FOR BANKRUPTCY

The first step was to determine if the individual had already been reported bankrupt. Bankrupt accounts would be indicated by comments starting with 'BK'. If an individual was found to be bankrupt then administrative action is left to the discretion of the investigating organization. Other less drastic but still serious situations, such as a bankruptcy petition filed or dismissed might also be indicated. In either case such an occurrence might warrant further investigation of this individual. [Ref. 1, pp. 8-10]

If no bankruptcy credit remarks were identified the second step was to check the credit report for the potential of a bankruptcy occurring. This was done by examining all credit remarks in the report, determining their overall weight value and then comparing that value to a previously established cut off level for potential bankruptcy. If the overall weight value exceeded that cut off level then a potential for bankruptcy was indicated. The rationale is that any individual accumulating a large number of derogatory accounts such as delinquent or collection accounts is placing himself in the position of facing potential bankruptcy.

Another indicator of potential bankruptcy is the number of inquiries shown in the credit report. An inquiry indicates an individual has approached a credit institution to borrow money. The rationale here is that if several inquiries appear in the report, this shows the individual was in desperate need of money and was trying hard to get the required credit by approaching as many institutions as he could. Almost always this characteristic is an indicator of an impending bankruptcy. [Ref. 1, pp. 10-12] The drawback to this rationale is the uncertainty of just how many inquiries constitute too many. For example, one individual might have over 30 inquiries in a given period if he was seeking credit for some legitimate, though semi-risky business venture he was considering trying. On the other hand another, more financially limited individual, desperate for cash, might have only approached up to eight credit institutions before he was able to prevail upon a credit authorizer that he deserved a loan. Since there was no sound heuristic available for determining when too many inquiries became a derogatory attribute, the inquiries were worked into the weighting scheme and calculated as part of the overall weight value. Figure 3.6 illustrates the flow of reasoning used by MFTES in bankruptcy analysis.

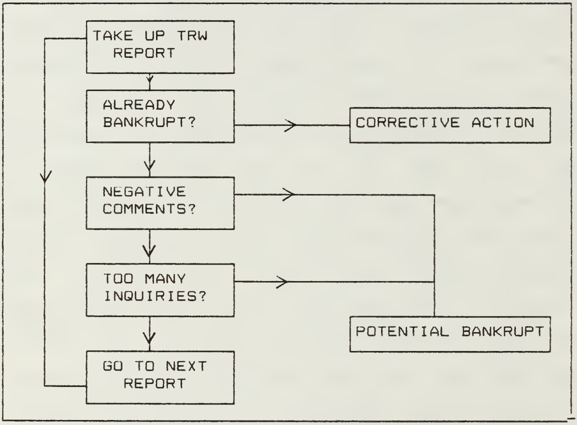

Figure 3.6 Flowchart for Bankruptcy Analysis

G. ANALYSIS FOR SUPERPAYER

If no potential bankruptcy was identified then the credit report was examined for a Superpayer condition. In this case an individual's discretionary monthly income was compared against the monthly payments shown in his TRW credit record. Monthly discretionary income was derived, as shown in Figure 3.7 by computing the employee's disposable income minus his expenses.

Disposable income was determined by first calculating an employee's annual income. The Annual Income Figure was obtained by comparing grade and step information to the General Schedule table which is completely modeled in MFTES.

Second a determination was made if the individual was married or not. If he was married then the Annual Income Figure was increased by 60 percent. This was a heuristic given by the financial expert as a reasonable increase in income that a working spouse (future program refinement must make a distinction between a working and non-working spouse), on average, will provide.

```
DISCRETIONARY INCOME = DISPOSABLE INCOME - EXPENSES

DISPOSABLE INCOME = ANNUAL INCOME FIGURE (AIF)
                      FROM GS SCHED

                    :IF MARRIED THEN AIF = AIF * 1.6

                    :DETERMINE TAXES
              IF AIF < 10,000

                        THEN TAX RATE = 0%
              IF AIF < 30,000

                        THEN TAX RATE = 15%
                     ELSE TAX RATE = 28%

                     NET RATE = 1 - TAX RATE

                    = AIF * NET RATE

  EXPENSES = RENT + FOOD
                    :RENT FROM ZIP CODE/RENT TABLE
             :FOOD = AIF * 20%


 MONTHLY DISCRETIONARY INCOME =
              DISPOSABLE INCOME - EXPENSES
                         12

MONTHLY PAYMENTS = TOTAL MONTHLY BALANCES DUE * 10%

   THEREFORE IF:

 MONTHLY PAYMENTS > MONTHLY DISCRETIONARY INCOME


       THEN    "SUPERPAYER"
```

Figure 3.7 Monthly Discretionary Income

Taxes also had to be computed. A Tax Rate, 0 percent if the annual income was less than $10,000 or 15 percent if the annual income was less than $30,000, was applied to the Annual Income Figure. The Net Rate was then figured by subtracting the Tax Rate from 1. The individual's disposable income was therefore calculated by multiplying the Annual Income Figure by the Net Rate.

Expenses were figured by calculating the individual's annual Rent plus his annual cost of Food. A more complete representation of expenses would take into account other outlays such as clothing, transportation, insurance, etc. but these were considered by the financial expert to have less impact in determining a Superpayer condition.

Annual Rent was computed by referencing a table of postal zip codes that were cross referenced to geographically established rents. In MFTES, a portion of this table was constructed for the Monterey--Naval Postgraduate School area to demonstrate its application in Rent computation. This information is not available on the TRW report and must be drawn from some other database (future program enhancement could have the expert system acquire data from multiple databases before analysis begins).

Food was calculated by multiplying the Annual Income Figure times 20 percent. The financial consultant provided this heuristic for the calculation of the average annual amount of money an American family spends on food. Monthly discretionary income then became Disposable Income minus Expenses divided by 12 months.

The employee's total monthly payment was computed from any Balances due that were found on the credit report. The Balances due for each account were summed together and then multiplied by 10%. This rule of thumb from the financial consultant asserts the average family's monthly payment on a balance due will not generally exceed 10 percent of that balance.

If the individual's monthly payments shown in his TRW credit record exceeded his discretionary monthly income then a superpayer condition existed. If no superpayer condition was determined then MFTES examined the individual's credit report for an overall weight value.

Another conclusion that can be made, and which infers a potentially detrimental situation, is the accumulation of too many current accounts. The presence of several current accounts connotes two possible situations. First, the individual might be inadvertently placing himself in the position of becoming overdrawn. This could happen if he suffers some type of financial setback and then is unable to fulfill payment obligations on his many accounts. Second, a large number of current accounts with low balances and high credit lines suggests the individual could use his combined total credit available to further extend himself into an unsound financial position. Thus the number of current accounts appearing in a credit record was worked into the weighting scheme and calculated as part of the overall weight value.

H. ANALYSIS OF OVERALL WEIGHT VALUE

If there was no indication of bankruptcy or superpayer then some other measure of an individual's financial position was needed. In order to fairly and accurately measure the effect of any possible combination of credit remarks appearing in an employee's credit record, an algorithm was needed that could handle the complexity of all combinatorial possibilities for the 102 different remarks. The weighting scheme for overall weight value that was used in the MFTES prototype, was chosen to be continued because of its ease of use, effectiveness and flexibility for future upgrade.

The overall weight value was determined by relating any remarks found in the report to a predefined point value (weight). These points were based a financial expert's best estimate. The points ranged on a scale from 10 to 200. Remarks with low derogatory value, such as a current account, paid account, inquiry, etc. received the least number of points (10) for each occurrence. This permitted such remarks to appear in a credit report without significantly impacting the overall weight value, although any large accumulation of these remarks would begin to affect it. Remarks with higher derogatory value, such as bankruptcy, judgments, liens and foreclosures received the most points (200) for each occurrence. This ensured that single remarks of this more serious type, which significantly affect an individual's financial position, had an immediate and heavy effect on the overall weight value.

Each occurrence of a remark was multiplied by its weight and all remarks were then summed together to produce the overall weight value. That overall weight value was then compared to a scale marked with different cutoff levels. The cutoff levels, also provided by the financial consultant, defined different financial positions. For example, Figure 3.8 (SSN has been altered) shows a sample case with two current accounts, a 120 day delinquent account and a collection account. The weights assigned to each remark and the current cut off levels are also shown. The weight assigned for each current account is 10, for the delinquent account is 40, and for the collection account is 200. The overall weight value then is 260 ((2 curr_accts * 10) + (1 delinq_120 * 40) + (1 coll_acct * 200)). A conclusion of Very Critical is assigned to any value that exceeds the Extremely High Cut Off Level of 200. If the overall weight value had exceeded 400 then a potential bankruptcy would have been indicated. This appears reasonable because it shows the employee, although supporting two current accounts, experiencing difficulty paying off other debts, a fact that might be significant to his future financial position. This also shows the emphasis (large weight value) given to the appearance of a collection account in the credit record.

```
employee(’555-55-5555’,2,9,single,’93943’).
data(_,curr_acct,386,285,r,186,2000,1749,86,0).
data(_,curr_acct,986,485,r,886,0,0,0,0).
data(_,delinq_120,686,685,r,486,1,478,360,1).
data(_,coll_acct,986,_,r,786,599,0,0,0).

weight_per_status(curr_acct,10).
weight_per_status(delinq_120,40).
weight_per_status(coll_acct,200).

PB_CUT_OFF_VAL is 400.
EXTREMELY_HIGH_CUT_OFF_VAL is 200.
VERY_HIGH_CUT_OFF_VAL is 170.
HIGH_CUT_OFF_VAL is 120.
MODERATELY_HIGH_CUT_OFF_VAL is 100.
LOW_CUT_OFF_VAL is 80.
EXTREMELY_LOW_CUT_OFF_VAL is 40.
VERY_LOW_CUT_OFF_VAL is 10.
```

Figure 3.8 Sample case with weights and cutoff levels

IV. EVALUATIONS, CONCLUSIONS AND RECOMMENDATIONS

A. EVALUATION RESULTS

During the evaluation phase of the expanded MFTES all cases that were tested were real credit reports provided by PERSEREC. Each case was manually translated into a usable format before being run through the expert system. Of the 75 cases evaluated, six required additional data input from the operator before MFTES was able to arrive at a conclusion. These cases, it was determined, had data improperly entered and were corrected before being re-evaluated. In all cases the resulting conclusion from MFTES matched the predetermined judgment of the testers. This duplicated the results of the 24 hypothetical cases that were used during the project's development.

Additional tests were required to more thoroughly evaluate the MFTES conclusions against those of a human financial expert not previously associated with the 75 test cases. An evaluation period with a financial consultant for this continued testing had been arranged but was not performed because of extenuating circumstances. Observations made during testing were:

- Some of the weights assigned to credit remarks were judged to be slightly higher than necessary and caused greater overall weights to show. This did not appear to appreciably change any MFTES conclusions but a 'fine-tuning' of the weights would make the system more accurate.

- The run-times experienced during the analysis of each case were observed to be ranging from 3 to as much as 23 minutes per case. The extreme was caused by three factors. First, the program involved considerable depth and breadth in its search strategies. Second, the Arity expert system shell, which had exceeded its limits in stack space several times during development, was probably reaching its limits once again. Third, the 8 Mhz microcomputer, on which MFTES was installed was too slow.

B. HARDWARE AND SOFTWARE REQUIREMENTS

The MFTES was initially developed and set up to run on an 8 Mhz IBM PC, or compatible, with 640K RAM and a hard disk (minimum 10 meg recommended for secondary storage). This was judged inefficient however because of the large size of the program and the inherent time delays that causes. In the latter stages of development a 3 megabyte virtual disk was used to run the program. MFTES consumed approximately 2.5 megs of the virtual memory available and was significantly faster. For example, the conclusion in a NORMAL case was reached in 3 to 5 minutes, while using virtual memory, and took 20 to 30 minutes without virtual memory. In order to obtain more efficient run times a 16-20 Mhz CPU is recommended.

The software required to work with the program (Arity Expert System) is available off the shelf. No special training requirements are considered necessary other than basic familiarization, depending on the user's computer literacy. Operation of MFTES is user friendly and driven by the interface.

C. SUCCESSES

The program was successful in incorporating all credit remarks used by TRW and in developing the necessary rules for producing a valid conclusion about an employee's financial position. In other words it successfully followed the documentary trail on an employee's credit report and made an inference about his potential risk.

A side benefit is that the program could also be used as a training aid for potential credit record analysts. The explanation facility's display of how conclusions were reached is an excellent learning tool.

D. LIMITATIONS AND FUTURE IMPROVEMENTS

The MFTES is limited to processing a single record at a time. This must be improved if the scrutiny of thousands of personnel holding security clearances has any chance of becoming efficient. A batch processing program that can review large numbers of records and 'kick' out those displaying normal attributes would be beneficial. Such a program could be used as a filter for the expert system, greatly reducing the number of reports that would be forwarded to MFTES for analysis. Additional rules can be added to improve the level of expertise provided by a financial expert.

Multiple complimentary types of expertise can be added to help provider broader analysis profiles. Examples include using medical and law enforcement expertise to develop psychological and criminal profiles. The expert system could provide conclusions specific to the type of profile desired or to an overall combination of all types available.

An interface can be developed that pipes TRW and DoD data directly into the expert system. This could be extended to cover any external database that the expert system required data from in order to complete its calculations. Examples include the grade and step information for monthly pay calculation and zip codes with their associated rental expenses for rent calculation. This kind of improvement would eliminate the extra step that now occurs in translating records to a more usable format before analysis.

The user interface, although straightforward and easy to use, is plain and could be improved by adding color, graphics and additional explanation features. An on-line help facility could further enhance use. Implementation of voice interaction through a Keytronics keyboard could facilitate an analyst's use of the expert system's explanation module. The analyst could then more naturally 'pose' the questions he wanted answered without having to worry about exact keyboard entry.

The expert system can have incorporated the ability to monitor TRW data longitudinally for the same subjects. This would provide a capability for comparisons of individuals and performing trend analysis.

The overall weight value calculation algorithm could be revised for improved accuracy and efficiency. Run-time efficiency as well as stack and storage problems might also be improved with a newer version of the Arity software.

Finally, more testing needs to be done. More real-world cases should be evaluated with different analysts for comparison. Only a thorough testing of MFTES would ensure weaknesses were properly identified. Possibly a benchmark program designed by financial analysts who currently perform credit record screening could be administered.

APPENDIX A - SOURCE CODE

This appendix contains the source code for the expanded MFTES. This includes the front end file (CRD13.ARI), the calculations file (CRD13.CAL), the rules file (CRD13.RUL), and the taxonomy file (CRD13.TAX). The source code was included because it was felt necessary to provide a hardcopy for documentation and maintenance, and to give anyone who wanted a convenient and complete way of examining the program's structure.

It is recognized that no one will probably input the code from this documentation. Arrangements can be made to obtain a copy of the software through Prof. Sivasankaran or Lcdr. Salazar at the Naval Postgraduate School.

```
          /* CRD13.ARI  (FRONT END FILE) */

/* This section of code starts the record reading
process.  Once a record has been analyzed the
information is discarded in favor of another record of
exit. */

start
   :-     nl, repeat,
   cls,
   tmove(5,10),
   wc(50,42),
   tmove(15,10),
   wc(50,42),
   tmove(10,10),
   put(7),put(7),
   write(' Input file to be interpreted: '),
   read(Name_of_file),
asserta(current_case(Name_of_file)),
[Name_of_file],
     [! run_once !],
 write($Would you like another consultation  ?$),
     read(X), gc(full),
     abolish(data/10),abolish(inq/2),
     abolish(employee_in_process/4),
     abolish(current_case/1),
     abolish(balance_due/1),
     abolish(total_monthly_expenses/1),
     nl,  X \= yes , X \=  y,
   cls,
   tmove(8,33),
   wc(15,42),
   tmove(12,33),
   wc(15,42),
   tmove(10,35),
   put(7),put(7),
   write(': Quitting!'),
  tmove(20,0).


run_once
   :-
   cls,
   tmove(5,10),
   wc(50,42),
   tmove(15,10),
   wc(50,42),
   tmove(10,10),
   wa(79,137),
   put(7),put(7),put(7),put(7),
   write('                 W O R K I N G ! ! ! '),
   tmove(17,10),
/*   read(Y),   */
   root_instance(perserec,I,N),
   eval(concluded,conclusion_info,I,Val,true,CF),
   fail.
run_once :-  nl.



             /* CRD13.CAL (CALCULATIONS) */

/* This section of code includes computations as layed
out in the rules. Each module retrieves a rule-
specified piece of data for calculation. */


get_ssn(I,personal,ssn,[SSN/1.0]) :-
     employee(SSN,GS,SAL_STEP,M_STATUS,ZIP),gc(full).

/*  This module for future use in requesting records by
SSN.....

write('Input Social Security of Employee to be
screened:'),
wca(9,'#,74),put(7),
   read(SSN),
   employee(SSN,GS,SAL_STEP,M_STATUS,ZIP),
   assert(employee(SSN,GS,SAL_STEP,M_STATUS,ZIP)).  */


get_m_status(I,personal,m_status,[M_STATUS/1.0]):-
    employee(SSN,GS,SAL_STEP,M_STATUS,ZIP).


get_zip(I,personal,zip,[ZIP/1.0]) :-
    employee(SSN,GS,SAL_STEP,M_STATUS,ZIP).


/*  The 1988 General Schedule for federal employees */

gs_sal_table(1,1,9811).
gs_sal_table(1,2,10139).
gs_sal_table(1,3,10465).
gs_sal_table(1,4,10791).
gs_sal_table(1,5,11117).
gs_sal_table(1,6,11309).
gs_sal_table(1,7,11631).
gs_sal_table(1,8,11955).
gs_sal_table(1,9,11970).
gs_sal_table(1,10,12275).
gs_sal_table(2,1,11032).
gs_sal_table(2,2,11294).
gs_sal_table(2,3,11659).
gs_sal_table(2,4,11970).
gs_sal_table(2,5,12103).
gs_sal_table(2,6,12459).
gs_sal_table(2,7,12815).
gs_sal_table(2,8,13171).
gs_sal_table(2,9,13527).
gs_sal_table(2,10,13883).
gs_sal_table(3,1,12038).
gs_sal_table(3,2,12439).
gs_sal_table(3,3,12840).
gs_sal_table(3,4,13241).
gs_sal_table(3,5,13642).
gs_sal_table(3,6,14043).
gs_sal_table(3,7,14444).
gs_sal_table(3,8,14845).
gs_sal_table(3,9,15246).
gs_sal_table(3,10,15647).
gs_sal_table(4,1,13513).
gs_sal_table(4,2,13963).
gs_sal_table(4,3,14413).
gs_sal_table(4,4,14863).
gs_sal_table(4,5,15313).
gs_sal_table(4,6,15763).
gs_sal_table(4,7,16213).
gs_sal_table(4,8,16663).
gs_sal_table(4,9,17113).
gs_sal_table(4,10,17563).
gs_sal_table(5,1,15118).
gs_sal_table(5,2,15622).
gs_sal_table(5,3,16126).
gs_sal_table(5,4,16630).
gs_sal_table(5,5,17134).
gs_sal_table(5,6,17638).
gs_sal_table(5,7,18142).
gs_sal_table(5,8,18646).
gs_sal_table(5,9,19150).
gs_sal_table(5,10,19654).
gs_sal_table(6,1,16851).
gs_sal_table(6,2,17413).
gs_sal_table(6,3,17975).
gs_sal_table(6,4,18537).
gs_sal_table(6,5,19099).
gs_sal_table(6,6,19661).
gs_sal_table(6,7,20223).
gs_sal_table(6,8,20785).
gs_sal_table(6,9,21347).
gs_sal_table(6,10,21909).
gs_sal_table(7,1,18726).
gs_sal_table(7,2,19350).
gs_sal_table(7,3,19974).
gs_sal_table(7,4,20598).
gs_sal_table(7,5,21222).
gs_sal_table(7,6,21846).
gs_sal_table(7,7,22470).
gs_sal_table(7,8,23094).
gs_sal_table(7,9,23718).
gs_sal_table(7,10,24342).
gs_sal_table(8,1,20739).
gs_sal_table(8,2,21430).
gs_sal_table(8,3,22121).
gs_sal_table(8,4,22812).
gs_sal_table(8,5,23503).
gs_sal_table(8,6,24194).
gs_sal_table(8,7,24885).
gs_sal_table(8,8,25576).
gs_sal_table(8,9,26267).
gs_sal_table(8,10,26958).
gs_sal_table(9,1,22907).
gs_sal_table(9,2,23671).
gs_sal_table(9,3,24435).
gs_sal_table(9,4,25199).
gs_sal_table(9,5,25963).
gs_sal_table(9,6,267727).
gs_sal_table(9,7,27491).
gs_sal_table(9,8,28255).
gs_sal_table(9,9,29091).
gs_sal_table(9,10,29783).
gs_sal_table(10,1,25226).
gs_sal_table(10,2,26067).
gs_sal_table(10,3,26908).
gs_sal_table(10,4,27749).
gs_sal_table(10,5,28590).
gs_sal_table(10,6,29431).
gs_sal_table(10,7,30272).
gs_sal_table(10,8,31113).
gs_sal_table(10,9,31954).
gs_sal_table(10,10,32795).
gs_sal_table(11,1,27716).
gs_sal_table(11,2,28640).
gs_sal_table(11,3,29564).
gs_sal_table(11,4,30488).
gs_sal_table(11,5,31412).
gs_sal_table(11,6,32336).
gs_sal_table(11,7,33260).
gs_sal_table(11,8,34184).
gs_sal_table(11,9,35108).
gs_sal_table(11,10,36032).
gs_sal_table(12,1,33218).
gs_sal_table(12,2,34325).
gs_sal_table(12,3,35432).
gs_sal_table(12,4,36539).
gs_sal_table(12,5,37646).
gs_sal_table(12,6,38753).
gs_sal_table(12,7,39860).
gs_sal_table(12,8,40967).
gs_sal_table(12,9,42074).
gs_sal_table(12,10,43181).
gs_sal_table(13,1,39501).
gs_sal_table(13,2,40818).
gs_sal_table(13,3,42135).
gs_sal_table(13,4,43452).
gs_sal_table(13,5,44769).
gs_sal_table(13,6,46086).
gs_sal_table(13,7,47403).
gs_sal_table(13,8,48720).
gs_sal_table(13,9,50037).
gs_sal_table(13,10,51354).
gs_sal_table(14,1,46679).
gs_sal_table(14,2,48235).
gs_sal_table(14,3,49791).
gs_sal_table(14,4,51347).
gs_sal_table(14,5,52903).
gs_sal_table(14,6,54459).
gs_sal_table(14,7,56015).
gs_sal_table(14,8,57571).
gs_sal_table(14,9,59127).
gs_sal_table(14,10,60683).
gs_sal_table(15,1,54907).
gs_sal_table(15,2,56737).
gs_sal_table(15,3,58567).
gs_sal_table(15,4,60397).
gs_sal_table(15,5,62227).
gs_sal_table(15,6,64057).
gs_sal_table(15,7,65887).
gs_sal_table(15,8,67717).
gs_sal_table(15,9,69547).
gs_sal_table(15,10,71377).
gs_sal_table(16,1,64397).
gs_sal_table(16,2,66544).
gs_sal_table(16,3,68691).
gs_sal_table(16,4,70838).
gs_sal_table(16,5,72500).
gs_sal_table(16,6,73660).
gs_sal_table(16,7,75765).
gs_sal_table(16,8,77870).
gs_sal_table(16,9,79975).
gs_sal_table(17,1,73958).
gs_sal_table(17,2,76423).
gs_sal_table(17,3,78888).
gs_sal_table(17,4,81353).
gs_sal_table(17,5,83818).
gs_sal_table(18,1,86682).


get_coll_acct_n(I,coll_acct,coll_acct_n,[COLL_AC-
CT_N/1.0]):-

findall(coll_acct,data(_,coll_acct,_,_,_,_,_,_,_,_),L),
  length(L,COLL_ACCT_N),gc(full).
get_d_60_n(I,d_60,d_60_n,[D_60_N/1.0]):-
findall(delinq_60,data(_,delinq_60,_,_,_,_,_,_,_,_),L),
     length(L,D_60_N),gc(full).

get_d_60_a(I,d_60,d_60_a,[D_60_A/1.0]):-
findall(Amount,data(_,delinq_60,_,_,_,_,_,Amount,_,_),-
A),
total_d_60_a(A,D_60_A),gc(full).
total_d_60_a([],0):- !.
total_d_60_a([A¦L],D_60_A):-
     total_d_60_a(L,Subtotal),
     D_60_A is Subtotal + A.

get_d_90_n(I,d_90,d_90_n,[D_90_N/1.0]):-
findall(delinq_90,data(_,delinq_90,_,_,_,_,_,_,_,_),L),
     length(L,D_90_N),gc(full).
get_d_90_a(I,d_90,d_90_a,[D_90_A/1.0]):-
findall(Amount,data(_,delinq_90,_,_,_,_,_,Amount,_,_),-
A),
total_d_90_a(A,D_90_A),gc(full).
total_d_90_a([],0):- !.
total_d_90_a([A¦L],D_90_A):-
     total_d_90_a(L,Subtotal),
     D_90_A is Subtotal + A.


get_d_120_n(I,d_120,d_120_n,[D_120_N/1.0]):-
findall(delinq_120,data(_,delinq_120,_,_,_,_,_,_,_,_),-
L),
      length(L,D_120_N),gc(full).
```

```
get_d_120_a(I,d_120,d_120_a,[D_120_A/1.0]):-
findall(Amount,data(_,delinq_120,_,_,_,_,_,Amount,_,_)-
,A),
total_d_120_a(A,D_120_A),gc(full).
total_d_120_a([],0):- !.
total_d_120_a([A|L],D_120_A):-
     total_d_120_a(L,Subtotal),
     D_120_A is Subtotal + A.

get_d_150_n(I,d_150,d_150_n,[D_150_N/1.0]):-
findall(delinq_150,data(_,delinq_150,_,_,_,_,_,_,_,_),-
L),
      length(L,D_150_N),gc(full).
get_d_150_a(I,d_150,d_150_a,[D_150_A/1.0]):-
findall(Amount,data(_,delinq_150,_,_,_,_,_,Amount,_,_)-
,A),
total_d_150_a(A,D_150_A),gc(full).
total_d_150_a([],0):- !.
total_d_150_a([A|L],D_150_A):-
     total_d_150_a(L,Subtotal),
     D_150_A is Subtotal + A.

get_d_180_n(I,d_180,d_180_n,[D_180_N/1.0]):-
findall(delinq_180,data(_,delinq_180,_,_,_,_,_,_,_,_),-
L),
      length(L,D_180_N),gc(full).
get_d_180_a(I,d_180,d_180_a,[D_180_A/1.0]):-
findall(Amount,data(_,delinq_180,_,_,_,_,_,Amount,_,_)-
,A),
total_d_180_a(A,D_180_A),gc(full).
total_d_180_a([],0):- !.
total_d_180_a([A|L],D_180_A):-
     total_d_180_a(L,Subtotal),
     D_180_A is Subtotal + A.

get_c_w_60_n(I,c_w_60,c_w_60_n,[C_w_60_N/1.0]):-
findall(c_w_60,data(_,c_w_60,_,_,_,_,_,_,_,_),L),
      length(L,C_w_60_N),gc(full).
get_c_w_60_a(I,c_w_60,c_w_60_a,[C_w_60_A/1.0]):-
findall(Amount,data(_,c_w_60,_,_,_,_,_,Amount,_,_),A),
total_c_w_60_a(A,C_w_60_A),gc(full).
total_c_w_60_a([],0):- !.
total_c_w_60_a([A|L],C_w_60_A):-
     total_c_w_60_a(L,Subtotal),
     C_w_60_A is Subtotal + A.

get_c_w_90_n(I,c_w_90,c_w_90_n,[C_w_90_N/1.0]):-
findall(c_w_90,data(_,c_w_90,_,_,_,_,_,_,_,_),L),
      length(L,C_w_90_N),gc(full).
```

```
get_c_w_90_a(I,c_w_90,c_w_90_a,[C_w_90_A/1.0]):-
findall(Amount,data(_,c_w_90,_,_,_,_,_,Amount,_,_),A),
total_c_w_90_a(A,C_w_90_A),gc(full).
total_c_w_90_a([],0):- !.
total_c_w_90_a([A¦L],C_w_90_A):-
     total_c_w_90_a(L,Subtotal),
     C_w_90_A is Subtotal + A.

get_c_w_120_n(I,c_w_120,c_w_120_n,[C_w_120_N/1.0]):-
findall(c_w_120,data(_,c_w_120,_,_,_,_,_,_,_,_),L),
        length(L,C_w_120_N),gc(full).
get_c_w_120_a(I,c_w_120,c_w_120_a,[C_w_120_A/1.0]):-
findall(Amount,data(_,c_w_120,_,_,_,_,_,Amount,_,_),A),
total_c_w_120_a(A,C_w_120_A),gc(full).
total_c_w_120_a([],0):- !.
total_c_w_120_a([A¦L],C_w_120_A):-
     total_c_w_120_a(L,Subtotal),
     C_w_120_A is Subtotal + A.

get_c_w_150_n(I,c_w_150,c_w_150_n,[C_w_150_N/1.0]):-
findall(c_w_150,data(_,c_w_150,_,_,_,_,_,_,_,_),L),
        length(L,C_w_150_N),gc(full).
get_c_w_150_a(I,c_w_150,c_w_150_a,[C_w_150_A/1.0]):-
findall(Amount,data(_,c_w_150,_,_,_,_,_,Amount,_,_),A),
total_c_w_150_a(A,C_w_150_A),gc(full).
total_c_w_150_a([],0):- !.
total_c_w_150_a([A¦L],C_w_150_A):-
     total_c_w_150_a(L,Subtotal),
     C_w_150_A is Subtotal + A.

get_c_w_180_n(I,c_w_180,c_w_180_n,[C_w_180_N/1.0]):-
findall(c_w_180,data(_,c_w_180,_,_,_,_,_,_,_,_),L),
        length(L,C_w_180_N),gc(full).
get_c_w_180_a(I,c_w_180,c_w_180_a,[C_w_180_A/1.0]):-
findall(Amount,data(_,c_w_180,_,_,_,_,_,Amount,_,_),A),
total_c_w_180_a(A,C_w_180_A),gc(full).
total_c_w_180_a([],0):- !.
total_c_w_180_a([A¦L],C_w_180_A):-
     total_c_w_180_a(L,Subtotal),
     C_w_180_A is Subtotal + A.

get_pdbydlr_n(I,pdbydlr,pdbydlr_n,[PDBYDLR_N/1.0]):-
findall(pdbydlr,data(_,pdbydlr,_,_,_,_,_,_,_,_),L),
  length(L,PDBYDLR_N),gc(full).
get_pdbydlr_a(I,pdbydlr,pdbydlr_a,[PDBYDLR_A/1.0]):-
findall(Amount,data(_,pdbydlr,_,_,_,_,_,Amount,_,_),A),
total_pdbydlr_a(A,PDBYDLR_A),gc(full).
total_pdbydlr_a([],0):- !.
total_pdbydlr_a([A¦L],PDBYDLR_A):-
```

```
     total_pdbydlr_a(L,Subtotal),
     PDBYDLR_A is Subtotal + A.

get_suit_n(I,suit,suit_n,[Suit_N/1.0]):-
findall(suit,data(_,suit,_,_,_,_,_,_,_),L),
     length(L,Suit_N),gc(full).
get_suit_a(I,suit,suit_a,[Suit_A/1.0]):-
findall(Amount,data(_,suit,_,_,_,_,Amount,_,_),A),
total_suit_a(A,Suit_A),gc(full).
total_suit_a([],0):- !.
total_suit_a([A¦L],Suit_A):-
     total_suit_a(L,Subtotal),
     Suit_A is Subtotal + A.

get_bk_vals(I,trw,bk_vals,[BK_VALS/1.0]):-
     bk_7_file(BK_7),
     bk_11_file(BK_11),
     bk_13_file(BK_13),
     bk_7_disc(BK_7_DISC),
     bk_7_dism(BK_7_DISM),
     bk_13_comp(COMP),
     BK_VALS is
BK_7+BK_11+BK_13+BK_7_DISC+BK_7_DISM+COMP,
gc(full).

bk_7_file(BK_7):-
     (data(_,bk_7_file,_,_,_,_,_,_,_),
     BK_7 is 200) ; BK_7 is 0.

bk_11_file(BK_11):-
     (data(_,bk_11_file,_,_,_,_,_,_,_),
     BK_11 is 200) ; BK_11 is 0.

bk_13_file(BK_13):-
     (data(_,bk_13_file,_,_,_,_,_,_,_),
     BK_13 is 200) ; BK_13 is 0.

bk_7_disc(BK_7_DISC):-
     (data(_,bk_7_disc,_,_,_,_,_,_,_),
     BK_7_DISC is 200) ; BK_7_DISC is 0.

bk_7_dism(BK_7_DISM):-
     (data(_,bk_7_dism,_,_,_,_,_,_,_),
     BK_7_DISM is 200) ; BK_7_DISM is 0.

bk_13_comp(COMP):-
     (data(_,bk_13_comp,_,_,_,_,_,_,_),
     COMP is 200) ; COMP is 0.
```

```
member(X,[X¦_]).

member(X,[_¦Y]):- member(X,Y).

show_bk_file_message(T_bk_vals,Bk_vals):-
nl,put(7),write('The person is bankrupt.'),
        bk_val_per_status(T_bk_vals,Bk_vals).


show_bk_disc_message(T_bk_vals,Bk_vals):-
        nl,put(7),
        write('The person is a discharged bankrupt.'),

   bk_val_per_status(T_bk_vals,Bk_vals).

show_bk_dism_message(T_bk_vals,Bk_vals):-
 nl,put(7),
write('The bankruptcy petition was
dismissed/completed.'),
bk_val_per_status(T_bk_vals,Bk_vals).

not_bankrupt_case(Bk_Vals):-
        bk_val_per_status(not_bankrupt,Bk_Vals).


/* Alternate method of searching for Bankrupt credit
remarks that may be modified in the future.

---- look_for_bk_remark(BK_VALS);BK_VALS is 0. ---

look_for_bk_remark(BK_VALS):-
        data(_,CREDITREMARK,_,_,_,_,_,_,_,_),
        case([CREDITREMARK=bk_7_file->VAL is 200,
                CREDITREMARK=bk_11_file->VAL is 200,
                CREDITREMARK=bk_13_file->VAL is 200,
                CREDITREMARK=bk_7_disc->VAL is 200,
                CREDITREMARK=bk_11_file->VAL is 200,
                CREDITREMARK=bk_7_dism->VAL is 200,
                CREDITREMARK=bk_11_file->VAL is 200,
                CREDITREMARK=bk_13_file->VAL is 200,
                CREDITREMARK=bk_13_comp->VAL is 200¦
                VAL is 0]),!,
        ifthenelse(VAL=200,BK_VALS is 200,next_bk_search).

next_bk_search:-
        data(_,CREDITREMARK,_,_,_,_,_,_,_,_),!,
        get_bk_vals(I,trw,bk_vals,[BK_VALS/1.0]).
      (  check_bk_file(Temp_BK_VALS,BK_VALS);
         check_bk_disc(Temp_BK_VALS,BK_VALS) );
      check_bk_dism(Temp_BK_VALS,BK_VALS) ).


get_bk_vals(I,trw,bk_vals,[BK_VALS/1.0]):-
     bk_val_per_status(not_bankrupt,BK_VALS).


check_bk_file(Temp_BK_VALS,BK_VALS):-
member(Temp_BK_VALS,[bk_7_file,bk_11_file,bk_13_file]),
show_bk_file_message(Temp_BK_VALS,BK_VALS).


check_bk_disc(Temp_BK_VALS,BK_VALS):-
     member(Temp_BK_VALS,[bk_7_disc,bk_11_disc]),
          show_bk_disc_message(Temp_BK_VALS,BK_VALS).

check_bk_dism(Temp_BK_VALS,BK_VALS):-
 member(Temp_BK_VALS,[bk_7_dism,bk_11_dism,bk_13_dism,
      bk_13_comp]),
     show_bk_dism_message(Temp_BK_VALS,BK_VALS).

*/


get_notpdaa_n(I,notpdaa,notpdaa_n,[Notpdaa_N/1.0]):-
findall(notpdaa,data(_,notpdaa,_,_,_,_,_,_,_),L),
length(L,Notpdaa_N),gc(full).
get_notpdaa_a(I,notpdaa,notpdaa_a,[Notpdaa_A/1.0]):-
findall(Amount,data(_,notpdaa,_,_,_,_,_,Amount,_,_),A),
total_notpdaa_a(A,Notpdaa_A),gc(full).
total_notpdaa_a([],0):- !.
total_notpdaa_a([A¦L],Notpdaa_A):-
     total_notpdaa_a(L,Subtotal),
     Notpdaa_A is Subtotal + A.

get_fclos_vals(I,foreclosure, foreclosure_weight_val,
[FCWTVAL/1.0]):-
     data(_,foreclosure,_,_,_,_,_,_,_),
     show_foreclosure_message,
     weight_per_status(foreclosure,FCWTVAL),gc(full).


get_fclos_vals(I,foreclosure,foreclosure_weight_val,
[FCWTVAL/1.0]):-
     FCWTVAL is 0,gc(full).

show_foreclosure_message:-
     nl,put(7),
 write('There is a foreclosure item in the report.').


get_judgment_vals(I,judgment,judgment_weight_val,
[JDWTVAL/1.0]):-
     data(_,judgment,_,_,_,_,_,_,_,_),
     show_judgment_message,
     weight_per_status(judgment,JDWTVAL),gc(full).

get_judgment_vals(I,judgment,judgment_weight_val,
[JDWTVAL/1.0]):-
     JDWTVAL is 0,gc(full).

show_judgment_message:-
     nl,put(7),
     write('There is a judgment item in the report.').


get_repo_vals(I,repo,repo_weight_val,
[REPOWTVAL/1.0]):-
     data(_,repo,_,_,_,_,_,_,_,_),
     show_repo_message,
     weight_per_status(repo,REPOWTVAL),gc(full).


get_repo_vals(I,repo,repo_weight_val,
[REPOWTVAL/1.0]):-
     REPOWTVAL is 0,gc(full).

show_repo_message:-
     nl,put(7),
     write('There is a repossessed item in the
report.').


reverse_text(Text):-
     name(Text,L),
     length(L,N),
     wa(N,112),
     put(7),
     write(Text).

get_pd_coll_ac_n(I,pd_coll_ac,pd_coll_ac_n,
[PD_COLL_AC_N/1.0]):-
findall(pd_coll_ac,data(_,pd_coll_ac,_,_,_,_,_,_,_,_),-
L),
length(L,PD_COLL_AC_N),gc(full).
```

```
get_charge_off_n(I,charge_off,charge_off_n,
[CHARGE_OFF_N/1.0]):-
findall(charge_off,data(_,charge_off,_,_,_,_,_,_,_),-
L),
 length(L,CHARGE_OFF_N),gc(full).

check_recent_inq(I,inquiry,no_of_inq,[NoE/1.0]):-
findall(DoI,inq(_,DoI),L),
     length(L,NoE_1),
     NoE_2 is NoE_1 - 10,
     ifthenelse(NoE_2 <0,NoE_3 is 0,NoE_3 is NoE_2),
   ifthen(NoE_3>5,show_inq_message),
     ifthenelse(NoE_3=0,NoE is 0,NoE is NoE_1).


show_inq_message:-
     nl, put(7),write('Too many inquiries').

get_curr_acct_n(I,superpayer,curr_acct_n,[N/1.0]):-
findall(curr_acct,data(_,curr_acct,_,_,_,_,_,_,_),L),
   length(L,N_1),
     N_2 is N_1 - 10,    /*only over 10 curr_accts are
bad */
 ifthenelse(N_2 < 0,N_3 is 0,N_3 is N_2),
     ifthenelse(N_3=0,N is 0,N is N_2),
     gc(full).

show_curr_acct_message:-
     nl,put(7),write('Too many active creditors').


/* compute_total_monthly_payment(111111116,5864). */
compute_total_monthly_payment(SSN,Expenses):-
     next_match,gc(full),
     total_monthly_expenses(Expenses),
     current_case(Current_Case),
[Current_Case],gc(full).
 /* This is to reintroduce the data() predicates just
wiped out during the total monthly payment computations

*/

next_match:-
     next_payment ; total_monthly_payments.

next_payment:-
    data(A,B,C,D,E,F,G,Balance_Due,H,I),!,
     ifthen(var(Balance_Due),Balance_Due is 0),
     asserta(balance_due(Balance_Due)),
retractall(data(A,B,C,D,E,F,G,Balance_Due,H,I)),gc(ful-
l),
  next_match.

total_monthly_payments:-
findall(Balance_Due,balance_due(Balance_Due),L),
total_balances(L,Total_Balance),
Expenses is Total_Balance*0.10,
assertz(total_monthly_expenses(Expenses)).

total_balances([],0):- !.
total_balances([S|L],Total_Balance):-
     total_balances(L,Subtotal),
     Total_Balance is Subtotal + S.


retractall(X):- retract(X),fail.
retractall(X):-retract((x:-y)),fail.
retractall(_).


compute_total_monthly_payment(_,0).
/*
compute_discretionary_income(111111116,single,93943,-
3000). */
/* compute_discretionary_income(_,_,_,3000). */

compute_discretionary_income(SSN,Discretionary_In-
come):-
     employee(SSN,GS_Level,Sal_Step,M_Status,Zip),
     gs_sal_table(GS_Level,Sal_Step,I),
     ifthenelse(M_Status=married, Income is I*1.60,
Income is I),
     tax(Income,Tax_rate),
     rent(Zip,Rent),
     Food is Income*0.20,
     Net_rate is 1-Tax_rate,
     Disposable_income is Income*Net_rate,
     Expenses is Rent+Food,
     Yrly_discretion_amount is
Disposable_income-Expenses,
 Discr_Inc is Yrly_discretion_amount/12,
     Discretionary_Income is
round(Discr_Inc,0),gc(full).

tax(Income,Tax_rate):-
     case([Income<10000->Tax_rate is 0,
          Income<30000->Tax_rate is 0.15|
          Tax_rate is 0.28]).
```

```
rent('93943',700).
rent('93949',700).


/*  All conclusions are based these weighted decisions.
*/

weighted_decision(10,normal).
weighted_decision(20,normal).
weighted_decision(30,normal).
weighted_decision(40,satisfactory).
weighted_decision(50,satisfactory).
weighted_decision(60,less_than_satisfactory).
weighted_decision(70,less_than_satisfactory).
weighted_decision(80,poor).
weighted_decision(90,poor).
weighted_decision(100,very_poor).
weighted_decision(110,very_poor).
weighted_decision(120,serious).
weighted_decision(130,serious).
weighted_decision(140,serious).
weighted_decision(150,very_serious).
weighted_decision(160,very_serious).
weighted_decision(170,critical).
weighted_decision(180,critical).
weighted_decision(190,critical).
weighted_decision(_,very_critical).



weight_per_status(pdbydlr,50).
weight_per_status(notpdaa,30).
weight_per_status(coll_acct,200).
weight_per_status(pd_coll_ac,30).
weight_per_status(charge_off,200).
weight_per_status(inquiry,10).
weight_per_status(curr_acct,10).
weight_per_status(foreclosure,200).
weight_per_status(judgment,200).
weight_per_status(repo,200).
weight_per_status(suit,70).
weight_per_status(delinq_60,30).
weight_per_status(delinq_90,30).
weight_per_status(delinq_120,40).
weight_per_status(delinq_150,40).
weight_per_status(delinq_180,50).
weight_per_status(c_w_60,10).
weight_per_status(c_w_90,10).
weight_per_status(c_w_120,20).
weight_per_status(c_w_150,20).
weight_per_status(c_w_180,30).
weight_per_status(insclaim,70).
weight_per_status(volnrepo,100).
weight_per_status(cwpd,10).
weight_per_status(cwpd_30,10).
weight_per_status(cwpd_30by2,20).
weight_per_status(cwpd_30by3,20).
weight_per_status(cwpd_30by4,30).
weight_per_status(cwpd_30by5,40).
weight_per_status(cwpd_30by6,40).
weight_per_status(accpd_30,20).
weight_per_status(accpd_30by2,20).
weight_per_status(accpd_30by3,20).
weight_per_status(accpd_30by4,30).
weight_per_status(accpd_30by5,40).
weight_per_status(accpd_30by6,40).
weight_per_status(pacc_wpd_30,10).
weight_per_status(pacc_wpd_30by23,20).
weight_per_status(pacc_wpd_30by4,30).
weight_per_status(pacc_wpd_30by5,40).
weight_per_status(pacc_wpd_30by6,40).
weight_per_status(pacc_wdel60,10).
weight_per_status(pacc_wdel90,20).
weight_per_status(pacc_wdel120,30).
weight_per_status(pacc_wdel150,40).
weight_per_status(pacc_wdel180,50).
weight_per_status(pd_repo,50).
weight_per_status(pd_chg_off,50).
weight_per_status(pd_foreclo,50).
weight_per_status(bkliqreo,200).
weight_per_status(settled,50).
weight_per_status(bk_adj_pln,200).
weight_per_status(scnl_nwloc,50).
weight_per_status(co_now_pay,50).
weight_per_status(fore_proc,200).
weight_per_status(gov_claim,70).
weight_per_status(close_np_aa,50).
weight_per_status(scnl,100).
weight_per_status(fed_tax_ln,200).
weight_per_status(fed_tax_rel,50).
weight_per_status(judgmt_sat,50).
weight_per_status(judg_vacat,10).
weight_per_status(mech_lien,200).
weight_per_status(mech_rele,50).
weight_per_status(mn_mtg_fil,10).
weight_per_status(nt_respon,20).
weight_per_status(stat_tx_ln,200).
weight_per_status(sta_tx_rel,50).
weight_per_status(suit_dismd,10).
weight_per_status(wage_asign,50).
weight_per_status(wa_release,10).
weight_per_status(refinanced,10).
weight_per_status(cr_cd_lost,10).
weight_per_status(clos_inac,10).
weight_per_status(transfered,10).
weight_per_status(too_new_rt,10).
weight_per_status(paid_satis,10).
weight_per_status(paid_acct,10).
weight_per_status(deceased,10).
weight_per_status(cr_ln_clos,50).
weight_per_status(redmd_repo,30).
weight_per_status(cur_was_col,30).
weight_per_status(cr_ln_rnst,10).
weight_per_status(cur_was_for,30).
weight_per_status(pd_not_aa,20).
weight_per_status(city_tx_ln,200).
weight_per_status(city_tx_rel,50).
weight_per_status(consel_ser,20).
weight_per_status(co_tax_ln,200).
weight_per_status(co_tax_rel,50).

weight_for_superpayer(200).

bk_val_per_status(not_bankrupt,0).
bk_val_per_status(bk_7_disc,200).
bk_val_per_status(bk_7_file,200).
bk_val_per_status(bk_11_file,200).
bk_val_per_status(bk_7_dism,200).
bk_val_per_status(bk_11_dism,200).
bk_val_per_status(bk_11_disc,200).
bk_val_per_status(bk_13_dism,200).
bk_val_per_status(bk_13_file,200).

bk_val_per_status(bk_13_comp,200).
get_insclaim_n(I,insclaim,insclaim_n,[Insclaim_N/1.0]-
):-
findall(insclaim,data(_,insclaim,_,_,_,_,_,_,_,_),L),
    length(L,Insclaim_N),gc(full).
get_insclaim_a(I,insclaim,insclaim_a,[Insclaim_A/1.0]-
):-
findall(Amount,data(_,insclaim,_,_,_,_,_,Amount,_,_),-
A),
total_insclaim_a(A,Insclaim_A),gc(full).
total_insclaim_a([],0):- !.
total_insclaim_a([A|L],Insclaim_A):-
     total_insclaim_a(L,Subtotal),
     Insclaim_A is Subtotal + A.

get_volnrepo_n(I,volnrepo,volnrepo_n,[Volnrepo_N/1.0]-
):-
findall(volnrepo,data(_,volnrepo,_,_,_,_,_,_,_),L),
   length(L,Volnrepo_N),gc(full).
get_volnrepo_a(I,volnrepo,volnrepo_a,[Volnrepo_A/1.0]-
):-
findall(Amount,data(_,volnrepo,_,_,_,_,Amount,_,_),-
A),
total_volnrepo_a(A,Volnrepo_A).
total_volnrepo_a([],0):- !.
total_volnrepo_a([A|L],Volnrepo_A):-
     total_volnrepo_a(L,Subtotal),
     Volnrepo_A is Subtotal + A.

get_cwpd_n(I,cwpd,cwpd_n,[Cwpd_N/1.0]):-
findall(cwpd,data(_,cwpd,_,_,_,_,_,_,_),L),
     length(L,Cwpd_N),gc(full).
get_cwpd_a(I,cwpd,cwpd_a,[Cwpd_A/1.0]):-
findall(Amount,data(_,cwpd,_,_,_,_,Amount,_,_),A),
total_cwpd_a(A,Cwpd_A),gc(full).
total_cwpd_a([],0):- !.
total_cwpd_a([A|L],Cwpd_A):-
     total_cwpd_a(L,Subtotal),
     Cwpd_A is Subtotal + A.

get_cwpd_30_n(I,cwpd_30,cwpd_30_n,[Cwpd_30_N/1.0]):-
findall(cwpd_30,data(_,cwpd_30,_,_,_,_,_,_,_),L),
 length(L,Cwpd_30_N),gc(full).
get_cwpd_30_a(I,cwpd_30,cwpd_30_a,[Cwpd_30_A/1.0]):-
findall(Amount,data(_,cwpd_30,_,_,_,_,Amount,_,_),A),
total_cwpd_30_a(A,Cwpd_30_A),gc(full).
total_cwpd_30_a([],0):- !.
total_cwpd_30_a([A|L],Cwpd_30_A):-
     total_cwpd_30_a(L,Subtotal),
     Cwpd_30_A is Subtotal + A.


get_cwpd_30by2_n(I,cwpd_30by2,cwpd_30by2_n,[Cwpd_30by2-
_N/1.0]):-
findall(cwpd_30by2,data(_,cwpd_30by2,_,_,_,_,_,_,_),-
L),
      length(L,Cwpd_30by2_N),gc(full).
get_cwpd_30by2_a(I,cwpd_30by2,cwpd_30by2_a,[Cwpd_30by2-
_A/1.0]):-
findall(Amount,data(_,cwpd_30by2,_,_,_,_,Amount,_,_)-
,A),
total_cwpd_30by2_a(A,Cwpd_30by2_A),gc(full).
```

```
total_cwpd_30by2_a([],0):- !.
total_cwpd_30by2_a([A|L],Cwpd_30by2_A):-
     total_cwpd_30by2_a(L,Subtotal),
     Cwpd_30by2_A is Subtotal + A.

get_cwpd_30by3_n(I,cwpd_30by3,cwpd_30by3_n,[Cwpd_30by3-
_N/1.0]):-
findall(cwpd_30by3,data(_,cwpd_30by3,_,_,_,_,_,_,_),-
L),
      length(L,Cwpd_30by3_N),gc(full).
get_cwpd_30by3_a(I,cwpd_30by3,cwpd_30by3_a,[Cwpd_30by3-
_A/1.0]):-
findall(Amount,data(_,cwpd_30by3,_,_,_,_,Amount,_,_)-
,A),
total_cwpd_30by3_a(A,Cwpd_30by3_A),gc(full).
total_cwpd_30by3_a([],0):- !.
total_cwpd_30by3_a([A|L],Cwpd_30by2_A):-
     total_cwpd_30by3_a(L,Subtotal),
     Cwpd_30by3_A is Subtotal + A.

get_cwpd_30by4_n(I,cwpd_30by4,cwpd_30by4_n,[Cwpd_30by4-
_N/1.0]):-
findall(cwpd_30by4,data(_,cwpd_30by4,_,_,_,_,_,_,_),-
L),
      length(L,Cwpd_30by4_N),gc(full).
get_cwpd_30by4_a(I,cwpd_30by4,cwpd_30by4_a,[Cwpd_30by4-
_A/1.0]):-
findall(Amount,data(_,cwpd_30by4,_,_,_,_,Amount,_,_)-
,A),
total_cwpd_30by4_a(A,Cwpd_30by4_A),gc(full).
total_cwpd_30by4_a([],0):- !.
total_cwpd_30by4_a([A|L],Cwpd_30by4_A):-
     total_cwpd_30by4_a(L,Subtotal),
     Cwpd_30by4_A is Subtotal + A.

get_cwpd_30by5_n(I,cwpd_30by5,cwpd_30by5_n,[Cwpd_30by5-
_N/1.0]):-
findall(cwpd_30by5,data(_,cwpd_30by5,_,_,_,_,_,_,_),-
L),
      length(L,Cwpd_30by5_N),gc(full).
get_cwpd_30by5_a(I,cwpd_30by5,cwpd_30by5_a,[Cwpd_30by5-
_A/1.0]):-
findall(Amount,data(_,cwpd_30by5,_,_,_,_,Amount,_,_)-
,A),
total_cwpd_30by5_a(A,Cwpd_30by5_A),gc(full).
total_cwpd_30by5_a([],0):- !.
total_cwpd_30by5_a([A|L],Cwpd_30by5_A):-
     total_cwpd_30by5_a(L,Subtotal),
     Cwpd_30by5_A is Subtotal + A.
```

```
get_cwpd_30by6_n(I,cwpd_30by6,cwpd_30by6_n,[Cwpd_30by6-
_N/1.0]):-
findall(cwpd_30by6,data(_,cwpd_30by6,_,_,_,_,_,_,_),-
L),
      length(L,Cwpd_30by6_N),gc(full).
get_cwpd_30by6_a(I,cwpd_30by6,cwpd_30by6_a,[Cwpd_30by6-
_A/1.0]):-
findall(Amount,data(_,cwpd_30by6,_,_,_,_,_,Amount,_,_)-
,A),
total_cwpd_30by6_a(A,Cwpd_30by6_A),gc(full).
total_cwpd_30by6_a([],0):- !.
total_cwpd_30by6_a([A|L],Cwpd_30by6_A):-
     total_cwpd_30by6_a(L,Subtotal),
     Cwpd_30by6_A is Subtotal + A.

get_accpd_30_n(I,accpd_30,accpd_30_n,[Accpd_30_N/1.0]-
):-
findall(accpd_30,data(_,accpd_30,_,_,_,_,_,_,_),L),
   length(L,Accpd_30_N),gc(full).
get_accpd_30_a(I,accpd_30,accpd_30_a,[Accpd_30_A/1.0]-
):-
findall(Amount,data(_,accpd_30,_,_,_,_,_,Amount,_,_),-
A),
total_accpd_30_a(A,Accpd_30_A),gc(full).
total_accpd_30_a([],0):- !.
total_accpd_30_a([A|L],Accpd_30_A):-
     total_accpd_30_a(L,Subtotal),
     Accpd_30_A is Subtotal + A.


get_accpd_30by2_n(I,accpd_30by2,accpd_30by2_n,
[Accpd_30by2_N/1.0]):-
findall(accpd_30by2,data(_,accpd_30by2,_,_,_,_,_,_,_-
),L),
 length(L,Accpd_30by2_N),gc(full).
get_accpd_30by2_a(I,accpd_30by2,accpd_30by2_a,
[Accpd_30by2_A/1.0]):-
findall(Amount,data(_,accpd_30by2,_,_,_,_,_,Amount,_,_-
),A),
total_accpd_30by2_a(A,Accpd_30by2_A),gc(full).
total_accpd_30by2_a([],0):- !.
total_accpd_30by2_a([A|L],Accpd_30by2_A):-
     total_accpd_30by2_a(L,Subtotal),
     Accpd_30by2_A is Subtotal + A.

get_accpd_30by3_n(I,accpd_30by3,accpd_30by3_n,
[Accpd_30by3_N/1.0]):-
```

```
findall(accpd_30by3,data(_,accpd_30by3,_,_,_,_,_,_,_-
),L),
      length(L,Accpd_30by3_N),gc(full).
get_accpd_30by3_a(I,accpd_30by3,accpd_30by3_a,
[Accpd_30by3_A/1.0]):-
findall(Amount,data(_,accpd_30by3,_,_,_,_,Amount,_,_-
),A),
total_accpd_30by3_a(A,Accpd_30by3_A),gc(full).
total_accpd_30by3_a([],0):- !.
total_accpd_30by3_a([A|L],Accpd_30by2_A):-
     total_accpd_30by3_a(L,Subtotal),
     Accpd_30by3_A is Subtotal + A.


get_accpd_30by4_n(I,accpd_30by4,accpd_30by4_n,
[Accpd_30by4_N/1.0]):-
findall(accpd_30by4,data(_,accpd_30by4,_,_,_,_,_,_,_-
),L),
  length(L,Accpd_30by4_N),gc(full).
get_accpd_30by4_a(I,accpd_30by4,accpd_30by4_a,
[Accpd_30by4_A/1.0]):-
findall(Amount,data(_,accpd_30by4,_,_,_,_,Amount,_,_-
),A),
total_accpd_30by4_a(A,Accpd_30by4_A),gc(full).
total_accpd_30by4_a([],0):- !.
total_accpd_30by4_a([A|L],Accpd_30by4_A):-
     total_accpd_30by4_a(L,Subtotal),
     Accpd_30by4_A is Subtotal + A.

get_accpd_30by5_n(I,accpd_30by5,accpd_30by5_n,
[Accpd_30by5_N/1.0]):-
findall(accpd_30by5,data(_,accpd_30by5,_,_,_,_,_,_,_-
),L),
  length(L,Accpd_30by5_N),gc(full).
get_accpd_30by5_a(I,accpd_30by5,accpd_30by5_a,
[Accpd_30by5_A/1.0]):-
findall(Amount,data(_,accpd_30by5,_,_,_,_,Amount,_,_-
),A),
total_accpd_30by5_a(A,Accpd_30by5_A),gc(full).
total_accpd_30by5_a([],0):- !.
total_accpd_30by5_a([A|L],Accpd_30by5_A):-
     total_accpd_30by5_a(L,Subtotal),
     Accpd_30by5_A is Subtotal + A.

get_accpd_30by6_n(I,accpd_30by6,accpd_30by6_n,
[Accpd_30by6_N/1.0]):-
findall(accpd_30by6,data(_,accpd_30by6,_,_,_,_,_,_,_-
),L),
      length(L,Accpd_30by6_N),gc(full).
```

```
get_accpd_30by6_a(I,accpd_30by6,accpd_30by6_a,
[Accpd_30by6_A/1.0]):-
findall(Amount,data(_,accpd_30by6,_,_,_,_,_,Amount,_,_-
),A),
total_accpd_30by6_a(A,Accpd_30by6_A),gc(full).
total_accpd_30by6_a([],0):- !.
total_accpd_30by6_a([A¦L],Accpd_30by6_A):-
     total_accpd_30by6_a(L,Subtotal),
     Accpd_30by6_A is Subtotal + A.


get_pacc_wpd_30_n(I,pacc_wpd_30,pacc_wpd_30_n,
[Pacc_wpd_30_N/1.0]):-
findall(pacc_wpd_30,data(_,pacc_wpd_30,_,_,_,_,_,_,_-
),L),
     length(L,Pacc_wpd_30_N),gc(full).
get_pacc_wpd_30_a(I,pacc_wpd_30,pacc_wpd_30_a,
[Pacc_wpd_30_A/1.0]):-
findall(Amount,data(_,pacc_wpd_30,_,_,_,_,Amount,_,_-
),A),
total_pacc_wpd_30_a(A,Pacc_wpd_30_A),gc(full).
total_pacc_wpd_30_a([],0):- !.
total_pacc_wpd_30_a([A¦L],Pacc_wpd_30_A):-
     total_pacc_wpd_30_a(L,Subtotal),
     Pacc_wpd_30_A is Subtotal + A.

get_pacc_wpd_30by23_n(I,pacc_wpd_30by23,pacc_wpd_30by2-
3_n,
[Pacc_wpd_30by23_N/1.0]):-
findall(pacc_wpd_30by23,
data(_,pacc_wpd_30by23,_,_,_,_,_,_,_),L),
      length(L,Pacc_wpd_30by23_N),gc(full).
get_pacc_wpd_30by23_a(I,pacc_wpd_30by23,pacc_wpd_30by2-
3_a,
[Pacc_wpd_30by23_A/1.0]):-
findall(Amount,data(_,pacc_wpd_30by23,_,_,_,_,Amount-
,_,_),A),
total_pacc_wpd_30by23_a(A,Pacc_wpd_30by23_A),gc(full).

total_pacc_wpd_30by23_a([],0):- !.
total_pacc_wpd_30by23_a([A¦L],Pacc_wpd_30by23_A):-
total_pacc_wpd_30by23_a(L,Subtotal),
     Pacc_wpd_30by23_A is Subtotal + A.
get_pacc_wpd_30by4_n(I,pacc_wpd_30by4,pacc_wpd_30by4_n,
[Pacc_wpd_30by4_N/1.0]):-
findall(pacc_wpd_30by4,data(_,pacc_wpd_30by4,_,_,_,_,_-
,_,_,_),L),
      length(L,Pacc_wpd_30by4_N),gc(full).
get_pacc_wpd_30by4_a(I,pacc_wpd_30by4,pacc_wpd_30by4_a,
[Pacc_wpd_30by4_A/1.0]):-
findall(Amount,data(_,pacc_wpd_30by4,_,_,_,_,_,Amount,-
_,_),A),
total_pacc_wpd_30by4_a(A,Pacc_wpd_30by4_A),gc(full).
total_pacc_wpd_30by4_a([],0):- !.
total_pacc_wpd_30by4_a([A|L],Pacc_wpd_30by4_A):-
total_pacc_wpd_30by4_a(L,Subtotal),
     Pacc_wpd_30by4_A is Subtotal + A.

get_pacc_wpd_30by5_n(I,pacc_wpd_30by5,pacc_wpd_30by5_n,
[Pacc_wpd_30by5_N/1.0]):-
findall(pacc_wpd_30by5,data(_,pacc_wpd_30by5,_,_,_,_,_-
,_,_,_),L),
      length(L,Pacc_wpd_30by5_N),gc(full).
get_pacc_wpd_30by5_a(I,pacc_wpd_30by5,pacc_wpd_30by5_a,
[Pacc_wpd_30by5_A/1.0]):-
findall(Amount,data(_,pacc_wpd_30by5,_,_,_,_,_,Amount,-
_,_),A),
total_pacc_wpd_30by5_a(A,Pacc_wpd_30by5_A),gc(full).
total_pacc_wpd_30by5_a([],0):- !.
total_pacc_wpd_30by5_a([A|L],Pacc_wpd_30by5_A):-
total_pacc_wpd_30by5_a(L,Subtotal),
     Pacc_wpd_30by5_A is Subtotal + A.

get_pacc_wpd_30by6_n(I,pacc_wpd_30by6,pacc_wpd_30by6_n,
[Pacc_wpd_30by6_N/1.0]):-
findall(pacc_wpd_30by6,data(_,pacc_wpd_30by6,_,_,_,_,_-
,_,_,_),L),
      length(L,Pacc_wpd_30by6_N),gc(full).
get_pacc_wpd_30by6_a(I,pacc_wpd_30by6,pacc_wpd_30by6_a,
[Pacc_wpd_30by6_A/1.0]):-
findall(Amount,data(_,pacc_wpd_30by6,_,_,_,_,_,Amount,-
_,_),A),
total_pacc_wpd_30by6_a(A,Pacc_wpd_30by6_A),gc(full).
total_pacc_wpd_30by6_a([],0):- !.
total_pacc_wpd_30by6_a([A|L],Pacc_wpd_30by6_A):-
total_pacc_wpd_30by6_a(L,Subtotal),
     Pacc_wpd_30by6_A is Subtotal + A.


get_pacc_wdel60_n(I;pacc_wdel60,pacc_wdel60_n,
[Pacc_wdel60_N/1.0]):-
findall(pacc_wdel60,data(_,pacc_wdel60,_,_,_,_,_,_,_,_-
),L),
 length(L,Pacc_wdel60_N),gc(full).
get_pacc_wdel60_a(I,pacc_wdel60,pacc_wdel60_a,
[Pacc_wdel60_A/1.0]):-
findall(Amount,data(_,pacc_wdel60,_,_,_,_,_,Amount,_,_-
),A),
total_pacc_wdel60_a(A,Pacc_wdel60_A),gc(full).
total_pacc_wdel60_a([],0):- !.
total_pacc_wdel60_a([A¦L],Pacc_wdel60_A):-
     total_pacc_wdel60_a(L,Subtotal),
     Pacc_wdel60_A is Subtotal + A.

get_pacc_wdel90_n(I,pacc_wdel90,pacc_wdel90_n,
[Pacc_wdel90_N/1.0]):-
findall(pacc_wdel90,data(_,pacc_wdel90,_,_,_,_,_,_,_,_-
),L),
      length(L,Pacc_wdel90_N),gc(full).
get_pacc_wdel90_a(I,pacc_wdel90,pacc_wdel90_a,
[Pacc_wdel90_A/1.0]):-
findall(Amount,data(_,pacc_wdel90,_,_,_,_,_,Amount,_,_-
),A),
total_pacc_wdel90_a(A,Pacc_wdel90_A),gc(full).
total_pacc_wdel90_a([],0):- !.
total_pacc_wdel90_a([A¦L],Pacc_wdel90_A):-
     total_pacc_wdel90_a(L,Subtotal),
     Pacc_wdel90_A is Subtotal + A.

get_pacc_wdel120_n(I,pacc_wdel120,pacc_wdel120_n,
[Pacc_wdel120_N/1.0]):-
findall(pacc_wdel120,data(_,pacc_wdel120,_,_,_,_,_,_,_-
,_),L),
    length(L,Pacc_wdel120_N),gc(full).
get_pacc_wdel120_a(I,pacc_wdel120,pacc_wdel120_a,
[Pacc_wdel120_A/1.0]):-
findall(Amount,data(_,pacc_wdel120,_,_,_,_,_,Amount,_,-
_),A),
total_pacc_wdel120_a(A,Pacc_wdel120_A),gc(full).
total_pacc_wdel120_a([],0):- !.
total_pacc_wdel120_a([A¦L],Pacc_wdel120_A):-
     total_pacc_wdel120_a(L,Subtotal),
     Pacc_wdel120_A is Subtotal + A.

get_pacc_wdel150_n(I,pacc_wdel150,pacc_wdel150_n,
[Pacc_wdel150_N/1.0]):-
findall(pacc_wdel150,data(_,pacc_wdel150,_,_,_,_,_,_,_-
,_),L),
    length(L,Pacc_wdel150_N),gc(full).
get_pacc_wdel150_a(I,pacc_wdel150,pacc_wdel150_a,
[Pacc_wdel150_A/1.0]):-
findall(Amount,data(_,pacc_wdel150,_,_,_,_,_,Amount,_,-
_),A),
total_pacc_wdel150_a(A,Pacc_wdel150_A),gc(full).
total_pacc_wdel150_a([],0):- !.
total_pacc_wdel150_a([A¦L],Pacc_wdel150_A):-
     total_pacc_wdel150_a(L,Subtotal),
     Pacc_wdel150_A is Subtotal + A.


get_pacc_wdel180_n(I,pacc_wdel180,pacc_wdel180_n,
[Pacc_wdel180_N/1.0]):-
findall(pacc_wdel180,data(_,pacc_wdel180,_,_,_,_,_,_,_-
,_),L),
    length(L,Pacc_wdel180_N),gc(full).
get_pacc_wdel180_a(I,pacc_wdel180,pacc_wdel180_a,
[Pacc_wdel180_A/1.0]):-
findall(Amount,data(_,pacc_wdel180,_,_,_,_,_,Amount,_,-
_),A),
total_pacc_wdel180_a(A,Pacc_wdel180_A),gc(full).
total_pacc_wdel180_a([],0):- !.
total_pacc_wdel180_a([A¦L],Pacc_wdel180_A):-
     total_pacc_wdel180_a(L,Subtotal),
     Pacc_wdel180_A is Subtotal + A.

get_pd_repo_n(I,pd_repo,pd_repo_n,
[Pd_repo_N/1.0]):-
findall(pd_repo,data(_,pd_repo,_,_,_,_,_,_,_,_),L),
      length(L,Pd_repo_N),gc(full).
get_pd_repo_a(I,pd_repo,pd_repo_a,
[Pd_repo_A/1.0]):-
findall(Amount,data(_,pd_repo,_,_,_,_,_,Amount,_,_),A),
total_pd_repo_a(A,Pd_repo_A),gc(full).
total_pd_repo_a([],0):- !.
total_pd_repo_a([A¦L],Pd_repo_A):-
     total_pd_repo_a(L,Subtotal),
     Pd_repo_A is Subtotal + A.

get_pd_chg_off_n(I,pd_chg_off,pd_chg_off_n,
[Pd_chg_off_N/1.0]):-
findall(pd_chg_off,data(_,pd_chg_off,_,_,_,_,_,_,_,_),-
L),
length(L,Pd_chg_off_N),gc(full).
get_pd_chg_off_a(I,pd_chg_off,pd_chg_off_a,
[Pd_chg_off_A/1.0]):-
findall(Amount,data(_,pd_chg_off,_,_,_,_,_,Amount,_,_)-
,A),
total_pd_chg_off_a(A,Pd_chg_off_A),gc(full).
total_pd_chg_off_a([],0):- !.
total_pd_chg_off_a([A¦L],Pd_chg_off_A):-
     total_pd_chg_off_a(L,Subtotal),
     Pd_chg_off_A is Subtotal + A.

get_pd_foreclo_n(I,pd_foreclo,pd_foreclo_n,
[Pd_foreclo_N/1.0]):-
findall(pd_foreclo,data(_,pd_foreclo,_,_,_,_,_,_,_),-
L),
length(L,Pd_foreclo_N),gc(full).
get_pd_foreclo_a(I,pd_foreclo,pd_foreclo_a,
[Pd_foreclo_A/1.0]):-
findall(Amount,data(_,pd_foreclo,_,_,_,_,_,Amount,_,_)-
,A),
total_pd_foreclo_a(A,Pd_foreclo_A),gc(full).
total_pd_foreclo_a([],0):- !.
total_pd_foreclo_a([A|L],Pd_foreclo_A):-
      total_pd_foreclo_a(L,Subtotal),
      Pd_foreclo_A is Subtotal + A.

get_bkliqreo_n(I,bkliqreo,bkliqreo_n,
[Bkliqreo_N/1.0]):-
findall(bkliqreo,data(_,bkliqreo,_,_,_,_,_,_,_),L),
       length(L,Bkliqreo_N),gc(full).
get_bkliqreo_a(I,bkliqreo,bkliqreo_a,
[Bkliqreo_A/1.0]):-
findall(Amount,data(_,bkliqreo,_,_,_,_,_,Amount,_,_),-
A),
total_bkliqreo_a(A,Bkliqreo_A),gc(full).
total_bkliqreo_a([],0):- !.
total_bkliqreo_a([A|L],Bkliqreo_A):-
      total_bkliqreo_a(L,Subtotal),
      Bkliqreo_A is Subtotal + A.


get_settled_n(I,settled,settled_n,
[Settled_N/1.0]):-
findall(settled,data(_,settled,_,_,_,_,_,_,_),L),
       length(L,Settled_N),gc(full).
get_settled_a(I,settled,settled_a,
[Settled_A/1.0]):-
findall(Amount,data(_,settled,_,_,_,_,_,Amount,_,_),A),
total_settled_a(A,Settled_A),gc(full).
total_settled_a([],0):- !.
total_settled_a([A|L],Settled_A):-
      total_settled_a(L,Subtotal),
      Settled_A is Subtotal + A.


get_bk_adj_pln_n(I,bk_adj_pln,bk_adj_pln_n,
[Bk_adj_pln_N/1.0]):-
findall(bk_adj_pln,data(_,bk_adj_pln,_,_,_,_,_,_,_),-
L),
length(L,Bk_adj_pln_N),gc(full).
get_bk_adj_pln_a(I,bk_adj_pln,bk_adj_pln_a,
[Bk_adj_pln_A/1.0]):-
findall(Amount,data(_,bk_adj_pln,_,_,_,_,_,Amount,_,_)-
,A),
total_bk_adj_pln_a(A,Bk_adj_pln_A),gc(full).
total_bk_adj_pln_a([],0):- !.
total_bk_adj_pln_a([A¦L],Bk_adj_pln_A):-
      total_bk_adj_pln_a(L,Subtotal),
      Bk_adj_pln_A is Subtotal + A.


get_scnl_nwloc_n(I,scnl_nwloc,scnl_nwloc_n,
[Scnl_nwloc_N/1.0]):-
findall(scnl_nwloc,data(_,scnl_nwloc,_,_,_,_,_,_,_),-
L),
length(L,Scnl_nwloc_N),gc(full).
get_scnl_nwloc_a(I,scnl_nwloc,scnl_nwloc_a,
[Scnl_nwloc_A/1.0]):-
findall(Amount,data(_,scnl_nwloc,_,_,_,_,_,Amount,_,_)-
,A),
total_scnl_nwloc_a(A,Scnl_nwloc_A),gc(full).
total_scnl_nwloc_a([],0):- !.
total_scnl_nwloc_a([A¦L],Scnl_nwloc_A):-
      total_scnl_nwloc_a(L,Subtotal),
      Scnl_nwloc_A is Subtotal + A.


get_co_now_pay_n(I,co_now_pay,co_now_pay_n,
[Co_now_pay_N/1.0]):-
findall(co_now_pay,data(_,co_now_pay,_,_,_,_,_,_,_),-
L),
length(L,Co_now_pay_N),gc(full).
get_co_now_pay_a(I,co_now_pay,co_now_pay_a,
[Co_now_pay_A/1.0]):-
findall(Amount,data(_,co_now_pay,_,_,_,_,_,Amount,_,_)-
,A),
total_co_now_pay_a(A,Co_now_pay_A),gc(full).
total_co_now_pay_a([],0):- !.
total_co_now_pay_a([A¦L],Co_now_pay_A):-
      total_co_now_pay_a(L,Subtotal),
      Co_now_pay_A is Subtotal + A.


get_fore_proc_n(I,fore_proc,fore_proc_n,
[Fore_proc_N/1.0]):-
findall(fore_proc,data(_,fore_proc,_,_,_,_,_,_,_),L),
       length(L,Fore_proc_N),gc(full).
get_fore_proc_a(I,fore_proc,fore_proc_a,
[Fore_proc_A/1.0]):-
```

```
findall(Amount,data(_,fore_proc,_,_,_,_,_,Amount,_,_),-
A),
total_fore_proc_a(A,Fore_proc_A),gc(full).
total_fore_proc_a([],0):- !.
total_fore_proc_a([A|L],Fore_proc_A):-
      total_fore_proc_a(L,Subtotal),
      Fore_proc_A is Subtotal + A.


get_gov_claim_n(I,gov_claim,gov_claim_n,
[Gov_claim_N/1.0]):-
findall(gov_claim,data(_,gov_claim,_,_,_,_,_,_,_,_),L),

       length(L,Gov_claim_N),gc(full).
get_gov_claim_a(I,gov_claim,gov_claim_a,
[Gov_claim_A/1.0]):-
findall(Amount,data(_,gov_claim,_,_,_,_,_,Amount,_,_),-
A),
total_gov_claim_a(A,Gov_claim_A),gc(full).
total_gov_claim_a([],0):- !.
total_gov_claim_a([A|L],Gov_claim_A):-
      total_gov_claim_a(L,Subtotal),
      Gov_claim_A is Subtotal + A.


get_close_np_aa_n(I,close_np_aa,close_np_aa_n,
[Close_np_aa_N/1.0]):-
findall(close_np_aa,data(_,close_np_aa,_,_,_,_,_,_,_,_-
),L),
 length(L,Close_np_aa_N),gc(full).
get_close_np_aa_a(I,close_np_aa,close_np_aa_a,
[Close_np_aa_A/1.0]):-
findall(Amount,data(_,close_np_aa,_,_,_,_,_,Amount,_,_-
),A),
total_close_np_aa_a(A,Close_np_aa_A),gc(full).
total_close_np_aa_a([],0):- !.
total_close_np_aa_a([A|L],Close_np_aa_A):-
      total_close_np_aa_a(L,Subtotal),
      Close_np_aa_A is Subtotal + A.


get_scnl_n(I,scnl,scnl_n,
[Scnl_N/1.0]):-
findall(scnl,data(_,scnl,_,_,_,_,_,_,_,_),L),
       length(L,Scnl_N),gc(full).
get_scnl_a(I,scnl,scnl_a,
[Scnl_A/1.0]):-
findall(Amount,data(_,scnl,_,_,_,_,_,Amount,_,_),A),
total_scnl_a(A,Scnl_A),gc(full).
total_scnl_a([],0):- !.
total_scnl_a([A|L],Scnl_A):-
     total_scnl_a(L,Subtotal),
     Scnl_A is Subtotal + A.


get_fed_tax_ln_n(I,fed_tax_ln,fed_tax_ln_n,
[Fed_tax_ln_N/1.0]):-
findall(fed_tax_ln,data(_,fed_tax_ln,_,_,_,_,_,_,_,_),-
L),
length(L,Fed_tax_ln_N),gc(full).
get_fed_tax_ln_a(I,fed_tax_ln,fed_tax_ln_a,
[Fed_tax_ln_A/1.0]):-
findall(Amount,data(_,fed_tax_ln,_,_,_,_,_,Amount,_,_)-
,A),
total_fed_tax_ln_a(A,Fed_tax_ln_A),gc(full).
total_fed_tax_ln_a([],0):- !.
total_fed_tax_ln_a([A|L],Fed_tax_ln_A):-
     total_fed_tax_ln_a(L,Subtotal),
     Fed_tax_ln_A is Subtotal + A.


get_fed_tax_rel_n(I,fed_tax_rel,fed_tax_rel_n,
[Fed_tax_rel_N/1.0]):-
findall(fed_tax_rel,data(_,fed_tax_rel,_,_,_,_,_,_,_,_-
),L),
 length(L,Fed_tax_rel_N),gc(full).
get_fed_tax_rel_a(I,fed_tax_rel,fed_tax_rel_a,
[Fed_tax_rel_A/1.0]):-
findall(Amount,data(_,fed_tax_rel,_,_,_,_,_,Amount,_,_-
),A),
total_fed_tax_rel_a(A,Fed_tax_rel_A),gc(full).
total_fed_tax_rel_a([],0):- !.
total_fed_tax_rel_a([A|L],Fed_tax_rel_A):-
     total_fed_tax_rel_a(L,Subtotal),
     Fed_tax_rel_A is Subtotal + A.


get_judgmt_sat_n(I,judgmt_sat,judgmt_sat_n,
[Judgmt_sat_N/1.0]):-
findall(judgmt_sat,data(_,judgmt_sat,_,_,_,_,_,_,_,_),-
L),
length(L,Judgmt_sat_N),gc(full).
get_judgmt_sat_a(I,judgmt_sat,judgmt_sat_a,
[Judgmt_sat_A/1.0]):-
findall(Amount,data(_,judgmt_sat,_,_,_,_,_,Amount,_,_)-
,A),
total_judgmt_sat_a(A,Judgmt_sat_A),gc(full).
total_judgmt_sat_a([],0):- !.
total_judgmt_sat_a([A¦L],Judgmt_sat_A):-
     total_judgmt_sat_a(L,Subtotal),
     Judgmt_sat_A is Subtotal + A.


get_judg_vacat_n(I,judg_vacat,judg_vacat_n,
[Judg_vacat_N/1.0]):-
findall(judg_vacat,data(_,judg_vacat,_,_,_,_,_,_,_),-
L),
length(L,Judg_vacat_N),gc(full).
get_judg_vacat_a(I,judg_vacat,judg_vacat_a,
[Judg_vacat_A/1.0]):-
findall(Amount,data(_,judg_vacat,_,_,_,_,Amount,_,_)-
,A),
total_judg_vacat_a(A,Judg_vacat_A),gc(full).
total_judg_vacat_a([],0):- !.
total_judg_vacat_a([A¦L],Judg_vacat_A):-
     total_judg_vacat_a(L,Subtotal),
     Judg_vacat_A is Subtotal + A.


get_mech_lien_n(I,mech_lien,mech_lien_n,
[Mech_lien_N/1.0]):-
findall(mech_lien,data(_,mech_lien,_,_,_,_,_,_,_),L),

      length(L,Mech_lien_N),gc(full).
get_mech_lien_a(I,mech_lien,mech_lien_a,
[Mech_lien_A/1.0]):-
findall(Amount,data(_,mech_lien,_,_,_,_,Amount,_,_),-
A),
total_mech_lien_a(A,Mech_lien_A),gc(full).
total_mech_lien_a([],0):- !.
total_mech_lien_a([A¦L],Mech_lien_A):-
     total_mech_lien_a(L,Subtotal),
     Mech_lien_A is Subtotal + A.


get_mech_rele_n(I,mech_rele,mech_rele_n,
[Mech_rele_N/1.0]):-
findall(mech_rele,data(_,mech_rele,_,_,_,_,_,_,_),L),
      length(L,Mech_rele_N),gc(full).
get_mech_rele_a(I,mech_rele,mech_rele_a,
[Mech_rele_A/1.0]):-
findall(Amount,data(_,mech_rele,_,_,_,_,Amount,_,_),-
A),
total_mech_rele_a(A,Mech_rele_A),gc(full).
total_mech_rele_a([],0):- !.
total_mech_rele_a([A¦L],Mech_rele_A):-
     total_mech_rele_a(L,Subtotal),
      Mech_rele_A is Subtotal + A.


get_mn_mtg_fil_n(I,mn_mtg_fil,mn_mtg_fil_n,
[Mn_mtg_fil_N/1.0]):-
findall(mn_mtg_fil,data(_,mn_mtg_fil,_,_,_,_,_,_,_,_),-
L),
length(L,Mn_mtg_fil_N),gc(full).
get_mn_mtg_fil_a(I,mn_mtg_fil,mn_mtg_fil_a,
[Mn_mtg_fil_A/1.0]):-
findall(Amount,data(_,mn_mtg_fil,_,_,_,_,_,Amount,_,_)-
,A),
total_mn_mtg_fil_a(A,Mn_mtg_fil_A),gc(full).
total_mn_mtg_fil_a([],0):- !.
total_mn_mtg_fil_a([A|L],Mn_mtg_fil_A):-
      total_mn_mtg_fil_a(L,Subtotal),
      Mn_mtg_fil_A is Subtotal + A.


get_nt_respon_n(I,nt_respon,nt_respon_n,
[Nt_respon_N/1.0]):-
findall(nt_respon,data(_,nt_respon,_,_,_,_,_,_,_,_),L),

       length(L,Nt_respon_N),gc(full).
get_nt_respon_a(I,nt_respon,nt_respon_a,
[Nt_respon_A/1.0]):-
findall(Amount,data(_,nt_respon,_,_,_,_,_,Amount,_,_),-
A),
total_nt_respon_a(A,Nt_respon_A),gc(full).
total_nt_respon_a([],0):- !.
total_nt_respon_a([A|L],Nt_respon_A):-
      total_nt_respon_a(L,Subtotal),
      Nt_respon_A is Subtotal + A.


get_stat_tx_ln_n(I,stat_tx_ln,stat_tx_ln_n,
[Stat_tx_ln_N/1.0]):-
findall(stat_tx_ln,data(_,stat_tx_ln,_,_,_,_,_,_,_,_),-
L),
length(L,Stat_tx_ln_N),gc(full).
get_stat_tx_ln_a(I,stat_tx_ln,stat_tx_ln_a,
[Stat_tx_ln_A/1.0]):-
findall(Amount,data(_,stat_tx_ln,_,_,_,_,_,Amount,_,_)-
,A),
total_stat_tx_ln_a(A,Stat_tx_ln_A),gc(full).
total_stat_tx_ln_a([],0):- !.
total_stat_tx_ln_a([A|L],Stat_tx_ln_A):-
      total_stat_tx_ln_a(L,Subtotal),
      Stat_tx_ln_A is Subtotal + A.
```

```
get_sta_tx_rel_n(I,sta_tx_rel,sta_tx_rel_n,
[Sta_tx_rel_N/1.0]):-
findall(sta_tx_rel,data(_,sta_tx_rel,_,_,_,_,_,_,_,_),-
L),
length(L,Sta_tx_rel_N),gc(full).
get_sta_tx_rel_a(I,sta_tx_rel,sta_tx_rel_a,
[Sta_tx_rel_A/1.0]):-
findall(Amount,data(_,sta_tx_rel,_,_,_,_,_,Amount,_,_)-
,A),
total_sta_tx_rel_a(A,Sta_tx_rel_A),gc(full).
total_sta_tx_rel_a([],0):- !.
total_sta_tx_rel_a([A|L],Sta_tx_rel_A):-
      total_sta_tx_rel_a(L,Subtotal),
      Sta_tx_rel_A is Subtotal + A.


get_suit_dismd_n(I,suit_dismd,suit_dismd_n,
[Suit_dismd_N/1.0]):-
findall(suit_dismd,data(_,suit_dismd,_,_,_,_,_,_,_,_),-
L),
length(L,Suit_dismd_N),gc(full).
get_suit_dismd_a(I,suit_dismd,suit_dismd_a,
[Suit_dismd_A/1.0]):-
findall(Amount,data(_,suit_dismd,_,_,_,_,_,Amount,_,_)-
,A),
total_suit_dismd_a(A,Suit_dismd_A),gc(full).
total_suit_dismd_a([],0):- !.
total_suit_dismd_a([A|L],Suit_dismd_A):-
      total_suit_dismd_a(L,Subtotal),
      Suit_dismd_A is Subtotal + A.


get_wage_asign_n(I,wage_asign,wage_asign_n,
[Wage_asign_N/1.0]):-
findall(wage_asign,data(_,wage_asign,_,_,_,_,_,_,_,_),-
L),
length(L,Wage_asign_N),gc(full).
get_wage_asign_a(I,wage_asign,wage_asign_a,
[Wage_asign_A/1.0]):-
findall(Amount,data(_,wage_asign,_,_,_,_,_,Amount,_,_)-
,A),
total_wage_asign_a(A,Wage_asign_A),gc(full).
total_wage_asign_a([],0):- !.
total_wage_asign_a([A|L],Wage_asign_A):-
      total_wage_asign_a(L,Subtotal),
      Wage_asign_A is Subtotal + A.
```

```
get_wa_release_n(I,wa_release,wa_release_n,
[Wa_release_N/1.0]):-
findall(wa_release,data(_,wa_release,_,_,_,_,_,_,_,_),-
L),
length(L,Wa_release_N),gc(full).
get_wa_release_a(I,wa_release,wa_release_a,
[Wa_release_A/1.0]):-
findall(Amount,data(_,wa_release,_,_,_,_,_,Amount,_,_)-
,A),
total_wa_release_a(A,Wa_release_A),gc(full).
total_wa_release_a([],0):- !.
total_wa_release_a([A|L],Wa_release_A):-
     total_wa_release_a(L,Subtotal),
     Wa_release_A is Subtotal + A.


get_refinanced_n(I,refinanced,refinanced_n,
[Refinanced_N/1.0]):-
findall(refinanced,data(_,refinanced,_,_,_,_,_,_,_,_),-
L),
length(L,Refinanced_N),gc(full).
get_refinanced_a(I,refinanced,refinanced_a,
[Refinanced_A/1.0]):-
findall(Amount,data(_,refinanced,_,_,_,_,_,Amount,_,_)-
,A),
total_refinanced_a(A,Refinanced_A),gc(full).
total_refinanced_a([],0):- !.
total_refinanced_a([A|L],Refinanced_A):-
     total_refinanced_a(L,Subtotal),
     Refinanced_A is Subtotal + A.


get_cr_cd_lost_n(I,cr_cd_lost,cr_cd_lost_n,
[Cr_cd_lost_N/1.0]):-
findall(cr_cd_lost,data(_,cr_cd_lost,_,_,_,_,_,_,_,_),-
L),
length(L,Cr_cd_lost_N),gc(full).
get_cr_cd_lost_a(I,cr_cd_lost,cr_cd_lost_a,
[Cr_cd_lost_A/1.0]):-
findall(Amount,data(_,cr_cd_lost,_,_,_,_,_,Amount,_,_)-
,A),
total_cr_cd_lost_a(A,Cr_cd_lost_A),gc(full).
total_cr_cd_lost_a([],0):- !.
total_cr_cd_lost_a([A|L],Cr_cd_lost_A):-
     total_cr_cd_lost_a(L,Subtotal),
     Cr_cd_lost_A is Subtotal + A.
```

```
get_clos_inac_n(I,clos_inac,clos_inac_n,
[Clos_inac_N/1.0]):-
findall(clos_inac,data(_,clos_inac,_,_,_,_,_,_,_),L),

      length(L,Clos_inac_N),gc(full).
get_clos_inac_a(I,clos_inac,clos_inac_a,
[Clos_inac_A/1.0]):-
findall(Amount,data(_,clos_inac,_,_,_,_,Amount,_,_),-
A),
total_clos_inac_a(A,Clos_inac_A),gc(full).
total_clos_inac_a([],0):- !.
total_clos_inac_a([A¦L],Clos_inac_A):-
     total_clos_inac_a(L,Subtotal),
     Clos_inac_A is Subtotal + A.


get_transfered_n(I,transfered,transfered_n,
[Transfered_N/1.0]):-
findall(transfered,data(_,transfered,_,_,_,_,_,_,_),-
L),
length(L,Transfered_N),gc(full).
get_transfered_a(I,transfered,transfered_a,
[Transfered_A/1.0]):-
findall(Amount,data(_,transfered,_,_,_,_,Amount,_,_)-
,A),
total_transfered_a(A,Transfered_A),gc(full).
total_transfered_a([],0):- !.
total_transfered_a([A¦L],Transfered_A):-
     total_transfered_a(L,Subtotal),
     Transfered_A is Subtotal + A.


get_too_new_rt_n(I,too_new_rt,too_new_rt_n,
[Too_new_rt_N/1.0]):-
findall(too_new_rt,data(_,too_new_rt,_,_,_,_,_,_,_),-
L),
length(L,Too_new_rt_N),gc(full).
get_too_new_rt_a(I,too_new_rt,too_new_rt_a,
[Too_new_rt_A/1.0]):-
findall(Amount,data(_,too_new_rt,_,_,_,_,Amount,_,_)-
,A),
total_too_new_rt_a(A,Too_new_rt_A),gc(full).
total_too_new_rt_a([],0):- !.
total_too_new_rt_a([A¦L],Too_new_rt_A):-
     total_too_new_rt_a(L,Subtotal),
     Too_new_rt_A is Subtotal + A.


get_paid_satis_n(I,paid_satis,paid_satis_n,
[Paid_satis_N/1.0]):-
findall(paid_satis,data(_,paid_satis,_,_,_,_,_,_,_),-
L),
length(L,Paid_satis_N),gc(full).
get_paid_satis_a(I,paid_satis,paid_satis_a,
[Paid_satis_A/1.0]):-
findall(Amount,data(_,paid_satis,_,_,_,_,Amount,_,_)-
,A),
total_paid_satis_a(A,Paid_satis_A),gc(full).
total_paid_satis_a([],0):- !.
total_paid_satis_a([A|L],Paid_satis_A):-
     total_paid_satis_a(L,Subtotal),
     Paid_satis_A is Subtotal + A.


get_paid_acct_n(I,paid_acct,paid_acct_n,
[Paid_acct_N/1.0]):-
findall(paid_acct,data(_,paid_acct,_,_,_,_,_,_,_),L),

      length(L,Paid_acct_N),gc(full).
get_paid_acct_a(I,paid_acct,paid_acct_a,
[Paid_acct_A/1.0]):-
findall(Amount,data(_,paid_acct,_,_,_,_,Amount,_,_),-
A),
total_paid_acct_a(A,Paid_acct_A),gc(full).
total_paid_acct_a([],0):- !.
total_paid_acct_a([A|L],Paid_acct_A):-
     total_paid_acct_a(L,Subtotal),
     Paid_acct_A is Subtotal + A.



get_deceased_n(I,deceased,deceased_n,
[Deceased_N/1.0]):-
findall(deceased,data(_,deceased,_,_,_,_,_,_,_),L),
      length(L,Deceased_N),gc(full).
get_deceased_a(I,deceased,deceased_a,
[Deceased_A/1.0]):-
findall(Amount,data(_,deceased,_,_,_,_,Amount,_,_),-
A),
total_deceased_a(A,Deceased_A),gc(full).
total_deceased_a([],0):- !.
total_deceased_a([A|L],Deceased_A):-
     total_deceased_a(L,Subtotal),
     Deceased_A is Subtotal + A.


get_cr_ln_clos_n(I,cr_ln_clos,cr_ln_clos_n,
[Cr_ln_clos_N/1.0]):-
```

```
findall(cr_ln_clos,data(_,cr_ln_clos,_,_,_,_,_,_,_),-
L),
length(L,Cr_ln_clos_N),gc(full).
get_cr_ln_clos_a(I,cr_ln_clos,cr_ln_clos_a,
[Cr_ln_clos_A/1.0]):-
findall(Amount,data(_,cr_ln_clos,_,_,_,_,Amount,_,_)-
,A),
total_cr_ln_clos_a(A,Cr_ln_clos_A),gc(full).
total_cr_ln_clos_a([],0):- !.
total_cr_ln_clos_a([A|L],Cr_ln_clos_A):-
     total_cr_ln_clos_a(L,Subtotal),
     Cr_ln_clos_A is Subtotal + A.


get_redmd_repo_n(I,redmd_repo,redmd_repo_n,
[Redmd_repo_N/1.0]):-
findall(redmd_repo,data(_,redmd_repo,_,_,_,_,_,_,_),-
L),
length(L,Redmd_repo_N),gc(full).
get_redmd_repo_a(I,redmd_repo,redmd_repo_a,
[Redmd_repo_A/1.0]):-
findall(Amount,data(_,redmd_repo,_,_,_,_,Amount,_,_)-
,A),
total_redmd_repo_a(A,Redmd_repo_A),gc(full).
total_redmd_repo_a([],0):- !.
total_redmd_repo_a([A|L],Redmd_repo_A):-
     total_redmd_repo_a(L,Subtotal),
     Redmd_repo_A is Subtotal + A.


get_cur_was_col_n(I,cur_was_col,cur_was_col_n,
[Cur_was_col_N/1.0]):-
findall(cur_was_col,data(_,cur_was_col,_,_,_,_,_,_,_-
),L),
 length(L,Cur_was_col_N),gc(full).
get_cur_was_col_a(I,cur_was_col,cur_was_col_a,
[Cur_was_col_A/1.0]):-
findall(Amount,data(_,cur_was_col,_,_,_,_,Amount,_,_-
),A),
total_cur_was_col_a(A,Cur_was_col_A),gc(full).
total_cur_was_col_a([],0):- !.
total_cur_was_col_a([A|L],Cur_was_col_A):-
     total_cur_was_col_a(L,Subtotal),
     Cur_was_col_A is Subtotal + A.


get_cr_ln_rnst_n(I,cr_ln_rnst,cr_ln_rnst_n,
[Cr_ln_rnst_N/1.0]):-
```

```
findall(cr_ln_rnst,data(_,cr_ln_rnst,_,_,_,_,_,_,_,_),-
L),
length(L,Cr_ln_rnst_N),gc(full).
get_cr_ln_rnst_a(I,cr_ln_rnst,cr_ln_rnst_a,
[Cr_ln_rnst_A/1.0]):-
findall(Amount,data(_,cr_ln_rnst,_,_,_,_,_,Amount,_,_)-
,A),
total_cr_ln_rnst_a(A,Cr_ln_rnst_A),gc(full).
total_cr_ln_rnst_a([],0):- !.
total_cr_ln_rnst_a([A|L],Cr_ln_rnst_A):-
     total_cr_ln_rnst_a(L,Subtotal),
     Cr_ln_rnst_A is Subtotal + A.


get_cur_was_for_n(I,cur_was_for,cur_was_for_n,
[Cur_was_for_N/1.0]):-
findall(cur_was_for,data(_,cur_was_for,_,_,_,_,_,_,_,_-
),L),
 length(L,Cur_was_for_N),gc(full).
get_cur_was_for_a(I,cur_was_for,cur_was_for_a,
[Cur_was_for_A/1.0]):-
findall(Amount,data(_,cur_was_for,_,_,_,_,_,Amount,_,_-
),A),
total_cur_was_for_a(A,Cur_was_for_A),gc(full).
total_cur_was_for_a([],0):- !.
total_cur_was_for_a([A|L],Cur_was_for_A):-
     total_cur_was_for_a(L,Subtotal),
     Cur_was_for_A is Subtotal + A.


get_pd_not_aa_n(I,pd_not_aa,pd_not_aa_n,
[Pd_not_aa_N/1.0]):-
findall(pd_not_aa,data(_,pd_not_aa,_,_,_,_,_,_,_,_),L),

      length(L,Pd_not_aa_N),gc(full).
get_pd_not_aa_a(I,pd_not_aa,pd_not_aa_a,
[Pd_not_aa_A/1.0]):-
findall(Amount,data(_,pd_not_aa,_,_,_,_,_,Amount,_,_),-
A),
total_pd_not_aa_a(A,Pd_not_aa_A),gc(full).
total_pd_not_aa_a([],0):- !.
total_pd_not_aa_a([A|L],Pd_not_aa_A):-
     total_pd_not_aa_a(L,Subtotal),
     Pd_not_aa_A is Subtotal + A.


get_city_tx_ln_n(I,city_tx_ln,city_tx_ln_n,
[City_tx_ln_N/1.0]):-
```

```
findall(city_tx_ln,data(_,city_tx_ln,_,_,_,_,_,_,_,_),-
L),
length(L,City_tx_ln_N),gc(full).
get_city_tx_ln_a(I,city_tx_ln,city_tx_ln_a,
[City_tx_ln_A/1.0]):-
findall(Amount,data(_,city_tx_ln,_,_,_,_,_,Amount,_,_)-
,A).
total_city_tx_ln_a(A,City_tx_ln_A),gc(full).
total_city_tx_ln_a([],0):- !.
total_city_tx_ln_a([A|L],City_tx_ln_A):-
     total_city_tx_ln_a(L,Subtotal),
     City_tx_ln_A is Subtotal + A.


get_city_tx_rel_n(I,city_tx_rel,city_tx_rel_n,
[City_tx_rel_N/1.0]):-
findall(city_tx_rel,data(_,city_tx_rel,_,_,_,_,_,_,_,_-
),L),
 length(L,City_tx_rel_N),gc(full).
get_city_tx_rel_a(I,city_tx_rel,city_tx_rel_a,
[City_tx_rel_A/1.0]):-
findall(Amount,data(_,city_tx_rel,_,_,_,_,_,Amount,_,_-
),A),
total_city_tx_rel_a(A,City_tx_rel_A),gc(full).
total_city_tx_rel_a([],0):- !.
total_city_tx_rel_a([A|L],City_tx_rel_A):-
     total_city_tx_rel_a(L,Subtotal),
     City_tx_rel_A is Subtotal + A.


get_consel_ser_n(I,consel_ser,consel_ser_n,
[Consel_ser_N/1.0]):-
findall(consel_ser,data(_,consel_ser,_,_,_,_,_,_,_,_),-
L),
length(L,Consel_ser_N),gc(full).
get_consel_ser_a(I,consel_ser,consel_ser_a,
[Consel_ser_A/1.0]):-
findall(Amount,data(_,consel_ser,_,_,_,_,_,Amount,_,_)-
,A),
total_consel_ser_a(A,Consel_ser_A),gc(full).
total_consel_ser_a([],0):- !.
total_consel_ser_a([A|L],Consel_ser_A):-
     total_consel_ser_a(L,Subtotal),
     Consel_ser_A is Subtotal + A.


get_co_tax_ln_n(I,co_tax_ln,co_tax_ln_n,
[Co_tax_ln_N/1.0]):-
```

```
findall(co_tax_ln,data(_,co_tax_ln,_,_,_,_,_,_,_,_),L),

      length(L,Co_tax_ln_N),gc(full).
get_co_tax_ln_a(I,co_tax_ln,co_tax_ln_a,
[Co_tax_ln_A/1.0]):-
findall(Amount,data(_,co_tax_ln,_,_,_,_,_,Amount,_,_),-
A),
total_co_tax_ln_a(A,Co_tax_ln_A),gc(full).
total_co_tax_ln_a([],0):- !.
total_co_tax_ln_a([A|L],Co_tax_ln_A):-
     total_co_tax_ln_a(L,Subtotal),
     Co_tax_ln_A is Subtotal + A.


get_co_tax_rel_n(I,co_tax_rel,co_tax_rel_n,
[Co_tax_rel_N/1.0]):-
findall(co_tax_rel,data(_,co_tax_rel,_,_,_,_,_,_,_,_),-
L),
length(L,Co_tax_rel_N),gc(full).
get_co_tax_rel_a(I,co_tax_rel,co_tax_rel_a,
[Co_tax_rel_A/1.0]):-
findall(Amount,data(_,co_tax_rel,_,_,_,_,_,Amount,_,_)-
,A),
total_co_tax_rel_a(A,Co_tax_rel_A),gc(full).
total_co_tax_rel_a([],0):- !.
total_co_tax_rel_a([A|L],Co_tax_rel_A):-
     total_co_tax_rel_a(L,Subtotal),
     Co_tax_rel_A is Subtotal + A.

get_pb_cut_off_val(I, cut_off,
pb_cut_off_val,[PB_CUT_OFF_VAL/1.0]):-
  PB_CUT_OFF_VAL is 400,gc(full).

get_extremely_high_cut_off_val(I, cut_off,
extremely_high_cut_off_val,
[EXTREMELY_HIGH_CUT_OFF_VAL/1.0]):-
     EXTREMELY_HIGH_CUT_OFF_VAL is 200,gc(full).

get_very_high_cut_off_val(I, cut_off,
very_high_cut_off_val,
[VERY_HIGH_CUT_OFF_VAL/1.0]):-
 VERY_HIGH_CUT_OFF_VAL is 170,gc(full).

get_high_cut_off_val(I, cut_off, high_cut_off_val,
[HIGH_CUT_OFF_VAL/1.0]):-
     HIGH_CUT_OFF_VAL is 120,gc(full).

get_moderately_high_cut_off_val(I, cut_off,
moderately_high_cut_off_val,
[MODERATELY_HIGH_CUT_OFF_VAL/1.0]):-
    MODERATELY_HIGH_CUT_OFF_VAL is 100,gc(full).

get_low_cut_off_val(I,
cut_off,low_cut_off_val,[LOW_CUT_OFF_VAL/1.0]):-
LOW_CUT_OFF_VAL is 80,gc(full).

get_very_low_cut_off_val(I,
cut_off,very_low_cut_off_val,
[VERY_LOW_CUT_OFF_VAL/1.0]):-
     VERY_LOW_CUT_OFF_VAL is 10,gc(full).

get_extremely_low_cut_off_val(I, cut_off,
extremely_low_cut_off_val,
[EXTREMELY_LOW_CUT_OFF_VAL/1.0]):-
EXTREMELY_LOW_CUT_OFF_VAL is 40,gc(full).



               /* CRD13.RUL  (RULES)  */

/*  This section of code contains synonyms and 90 if-
then rules */

report(concluded of conclusion) = $The person is :$.
order(concluded of conclusion) = [r,q].
precalc(personal) = [ssn,zip,m_status].
set(customer_profile_info of perserec) =
     customer_profile_info.
set(superpayer_info of perserec) = superpayer_info.
set(bankrupt_info of perserec) = bankrupt_info.
set(weight_pt_info of perserec) = weight_pt_info.
set(cut_off_info of perserec) = cut_off_info.

name(weight_pt_info of perserec) = ignore.
name(cut_off_info of perserec) = ignore.

question(pb_cut_off_val of cut_off) =
get_pb_cut_off_val.
question(extremely_high_cut_off_val of cut_off) =
     get_extremely_high_cut_off_val.
question(very_high_cut_off_val of cut_off) =
     get_very_high_cut_off_val.
question(high_cut_off_val of cut_off) =
     get_high_cut_off_val.
question(moderately_high_cut_off_val of cut_off) =
     get_moderately_high_cut_off_val.
```

```
question(low_cut_off_val of cut_off) =
     get_low_cut_off_val.
question(very_low_cut_off_val of cut_off) =
     get_very_low_cut_off_val.
question(extremely_low_cut_off_val of cut_off) =
     get_extremely_low_cut_off_val.

set(conclusion_info of perserec) =  conclusion_info.
set(personal_info of customer_profile_info of
    perserec)=
     personal_info.
set(trw_info of customer_profile_info of perserec) =
trw_info.
set(delinq_info of trw_info of customer_profile_info of
    perserec) = delinq_info.
set(d_60_info of delinq_info of trw_info of
     customer_profile_info of perserec) = d_60_info.
set(d_90_info of delinq_info of trw_info of
     customer_profile_info of perserec) = d_90_info.
set(d_120_info of delinq_info of trw_info of
     customer_profile_info of perserec) = d_120_info.
set(d_150_info of delinq_info of trw_info of
     customer_profile_info of perserec) = d_150_info.
set(d_180_info of delinq_info of trw_info of
     customer_profile_info of perserec) = d_180_info.
set(cur_was_info of trw_info of customer_profile_info
    of
     perserec) = cur_was_info.
set(c_w_60_info of cur_was_info of trw_info of
     customer_profile_info of perserec) = c_w_60_info.
set(c_w_90_info of cur_was_info of trw_info of
     customer_profile_info of perserec) = c_w_90_info.
set(c_w_120_info of cur_was_info of trw_info of
     customer_profile_info of perserec) = c_w_120_info.
set(c_w_150_info of cur_was_info of trw_info of
    customer_profile_info of perserec) = c_w_150_info.
set(c_w_180_info of cur_was_info of trw_info of
     customer_profile_info of perserec) = c_w_180_info.
set(pdbydlr_info of trw_info of
     customer_profile_info of perserec) = pdbydlr_info.
set(coll_acct_info of trw_info of customer_profile_info
    of
      perserec)= coll_acct_info.
set(charge_off_info of trw_info of
    customer_profile_info of
      perserec)=charge_off_info.
set(pd_coll_ac_info of trw_info of
    customer_profile_info of
      perserec) = pd_coll_ac_info.
```

```
set(inquiry_info of trw_info of customer_profile_info
    of
     perserec) = inquiry_info.
set(conclusion_info of perserec) = conclusion_info.
set(foreclosure_info of trw_info of
    customer_profile_info of
     perserec) = foreclosure_info.
set(judgment_info of trw_info of customer_profile_info
    of
     perserec) = judgment_info.
set(repo_info of trw_info of customer_profile_info of
    perserec) = repo_info.
set(insclaim_info of trw_info of
     customer_profile_info of perserec) =
    insclaim_info.

name(insclaim_info of trw) = ignore.
question(insclaim_n of insclaim) = get_insclaim_n.
question(insclaim_a of insclaim) = get_insclaim_a.


set(notpdaa_info of trw_info of
     customer_profile_info of perserec) = notpdaa_info.
name(notpdaa_info of trw) = ignore.
question(notpdaa_n of notpdaa) = get_notpdaa_n.
question(notpdaa_a of notpdaa) = get_notpdaa_a.

set(volnrepo_info of trw_info of
     customer_profile_info of perserec) =
volnrepo_info.
name(volnrepo_info of trw) = ignore.
question(volnrepo_n of volnrepo) = get_volnrepo_n.
question(volnrepo_a of volnrepo) = get_volnrepo_a.


set(curwaspd_info of trw_info of customer_profile_info
    of
    perserec) = curwaspd_info.
set(cwpd_info of curwaspd_info of trw_info of
     customer_profile_info of perserec) = cwpd_info.
set(cwpd_30_info of curwaspd_info of trw_info of
    customer_profile_info of perserec) = cwpd_30_info.


set(cwpd_30by2_info of curwaspd_info of trw_info of
    customer_profile_info of perserec) =
cwpd_30by2_info.
set(cwpd_30by3_info of curwaspd_info of trw_info of
    customer_profile_info of perserec) =
cwpd_30by3_info.
set(cwpd_30by4_info of curwaspd_info of trw_info of
    customer_profile_info of perserec) =
cwpd_30by4_info.
set(cwpd_30by5_info of curwaspd_info of trw_info of
customer_profile_info of perserec) = cwpd_30by5_info.
set(cwpd_30by6_info of curwaspd_info of trw_info of
customer_profile_info of perserec) = cwpd_30by6_info.

set(accpd_info of trw_info of customer_profile_info of
perserec) = accpd_info.
set(accpd_30_info of accpd_info of trw_info of
     customer_profile_info of perserec) =
accpd_30_info.

set(accpd_30by2_info of accpd_info of trw_info of
customer_profile_info of perserec) = accpd_30by2_info.
set(accpd_30by3_info of accpd_info of trw_info of
customer_profile_info of perserec) = accpd_30by3_info.
set(accpd_30by4_info of accpd_info of trw_info of
customer_profile_info of perserec) = accpd_30by4_info.
set(accpd_30by5_info of accpd_info of trw_info of
customer_profile_info of perserec) = accpd_30by5_info.
set(accpd_30by6_info of accpd_info of trw_info of
customer_profile_info of perserec) = accpd_30by6_info.
set(pacc_wpd_info of trw_info of customer_profile_info
    of
    perserec) = pacc_wpd_info.
set(pacc_wpd_30_info of pacc_wpd_info of trw_info of
 customer_profile_info of perserec) = pacc_wpd_30_info.
set(pacc_wpd_30by23_info of pacc_wpd_info of trw_info
    of
     customer_profile_info of perserec) =
pacc_wpd_30by23_info.
set(pacc_wpd_30by4_info of pacc_wpd_info of trw_info of
    customer_profile_info of perserec) =
pacc_wpd_30by4_info.
set(pacc_wpd_30by5_info of pacc_wpd_info of trw_info of
    customer_profile_info of perserec) =
pacc_wpd_30by5_info.
set(pacc_wpd_30by6_info of pacc_wpd_info of trw_info of
    customer_profile_info of perserec) =
pacc_wpd_30by6_info.

set(suit_info of trw_info of
     customer_profile_info of perserec) = suit_info.
name(suit_info of trw) = ignore.
question(suit_n of suit) = get_suit_n.
question(suit_a of suit) = get_suit_a.


name(customer_profile_info of perserec) = ignore.
name(superpayer_info of perserec) = ignore.
name(bankrupt_info of perserec) = ignore.
name(conclusion_info of perserec) = ignore.
name(personal_info of customer_profile) = ignore.
name(trw_info of customer_profile) = ignore.
name(inquiry_info of trw) = ignore.
name(delinq_info of trw) = ignore.

name(d_60_info of delinq) = ignore.
name(d_90_info of delinq) = ignore.
name(d_120_info of delinq) = ignore.
name(d_150_info of delinq) = ignore.
name(d_180_info of delinq) = ignore.

name(cur_was_info of trw) = ignore.
name(c_w_60_info of cur_was) = ignore.
name(c_w_90_info of cur_was) = ignore.
name(c_w_120_info of cur_was) = ignore.
name(c_w_150_info of cur_was) = ignore.
name(c_w_180_info of cur_was) = ignore.

name(coll_acct_info of trw) = ignore.
name(charge_off_info of trw) = ignore.
name(pd_coll_ac_info of trw) = ignore.
name(foreclosure_info of trw) = ignore.
name(judgment_info of trw) = ignore.
name(repo_info of trw) = ignore.
name(pdbydlr_info of trw) = ignore.
name(curwaspd_info of trw) = ignore.
name(cwpd_info of curwaspd) = ignore.
name(cwpd_30_info of curwaspd) = ignore.

name(cwpd_30by2_info of curwaspd) = ignore.
name(cwpd_30by3_info of curwaspd) = ignore.
name(cwpd_30by4_info of curwaspd) = ignore.
name(cwpd_30by5_info of curwaspd) = ignore.
name(cwpd_30by6_info of curwaspd) = ignore.

question(cwpd_n of cwpd) = get_cwpd_n.
question(cwpd_a of cwpd) = get_cwpd_a.
question(cwpd_30_n of cwpd_30) = get_cwpd_30_n.
question(cwpd_30_a of cwpd_30) = get_cwpd_30_a.
question(cwpd_30by2_n of cwpd_30by2) =
get_cwpd_30by2_n.
```

```
question(cwpd_30by2_a of cwpd_30by2) =
get_cwpd_30by2_a.
question(cwpd_30by3_n of cwpd_30by3) =
get_cwpd_30by3_n.
question(cwpd_30by3_a of cwpd_30by3) =
get_cwpd_30by3_a.
question(cwpd_30by4_n of cwpd_30by4) =
get_cwpd_30by4_n.
question(cwpd_30by4_a of cwpd_30by4) =
get_cwpd_30by4_a.
question(cwpd_30by5_n of cwpd_30by5) =
get_cwpd_30by5_n.
question(cwpd_30by5_a of cwpd_30by5) =
get_cwpd_30by5_a.
question(cwpd_30by6_n of cwpd_30by6) =
get_cwpd_30by6_n.
question(cwpd_30by6_a of cwpd_30by6) =
get_cwpd_30by6_a.

question(ssn of personal) = get_ssn.
question(m_status of personal) = get_m_status.
question(zip of personal) = get_zip.
question(coll_acct_n of coll_acct) = get_coll_acct_n.
question(coll_acct_a of coll_acct) = get_coll_acct_a.

question(c_w_60_n of c_w_60) = get_c_w_60_n.
question(c_w_60_a of c_w_60) = get_c_w_60_a.
question(c_w_90_n of c_w_90) = get_c_w_90_n.
question(c_w_90_a of c_w_90) = get_c_w_90_a.
question(c_w_120_n of c_w_120) = get_c_w_120_n.
question(c_w_120_a of c_w_120) = get_c_w_120_a.
question(c_w_150_n of c_w_150) = get_c_w_150_n.
question(c_w_150_a of c_w_150) = get_c_w_150_a.
question(c_w_180_n of c_w_180) = get_c_w_180_n.
question(c_w_180_a of c_w_180) = get_c_w_180_a.


question(d_60_n of d_60) = get_d_60_n.
question(d_60_a of d_60) = get_d_60_a.
question(d_90_n of d_90) = get_d_90_n.
question(d_90_a of d_90) = get_d_90_a.
question(d_120_n of d_120) = get_d_120_n.
question(d_120_a of d_120) = get_d_120_a.
question(d_150_n of d_150) = get_d_150_n.
question(d_150_a of d_150) = get_d_150_a.
question(d_180_n of d_180) = get_d_180_n.
question(d_180_a of d_180) = get_d_180_a.

name(accpd_info of trw) = ignore.
name(accpd_30_info of accpd) = ignore.
name(accpd_30by2_info of accpd) = ignore.
name(accpd_30by3_info of accpd) = ignore.
name(accpd_30by4_info of accpd) = ignore.
name(accpd_30by5_info of accpd) = ignore.
name(accpd_30by6_info of accpd) = ignore.

question(accpd_30_n of accpd_30) = get_accpd_30_n.
question(accpd_30_a of accpd_30) = get_accpd_30_a.
question(accpd_30by2_n of accpd_30by2) =
get_accpd_30by2_n.
question(accpd_30by2_a of accpd_30by2) =
get_accpd_30by2_a.
question(accpd_30by3_n of accpd_30by3) =
get_accpd_30by3_n.
question(accpd_30by3_a of accpd_30by3) =
get_accpd_30by3_a.
question(accpd_30by4_n of accpd_30by4) =
get_accpd_30by4_n.
question(accpd_30by4_a of accpd_30by4) =
get_accpd_30by4_a.
question(accpd_30by5_n of accpd_30by5) =
get_accpd_30by5_n.
question(accpd_30by5_a of accpd_30by5) =
get_accpd_30by5_a.
question(accpd_30by6_n of accpd_30by6) =
get_accpd_30by6_n.
question(accpd_30by6_a of accpd_30by6) =
get_accpd_30by6_a.

name(pacc_wpd_info of trw) = ignore.
name(pacc_wpd_30_info of pacc_wpd) = ignore.
name(pacc_wpd_30by23_info of pacc_wpd) = ignore.
name(pacc_wpd_30by4_info of pacc_wpd) = ignore.
name(pacc_wpd_30by5_info of pacc_wpd) = ignore.
name(pacc_wpd_30by6_info of pacc_wpd) = ignore.

question(pacc_wpd_30_n of pacc_wpd_30) =
get_pacc_wpd_30_n.
question(pacc_wpd_30_a of pacc_wpd_30) =
get_pacc_wpd_30_a.

question(pacc_wpd_30by23_n of pacc_wpd_30by23) =
  get_pacc_wpd_30by23_n.
question(pacc_wpd_30by23_a of pacc_wpd_30by23) =
  get_pacc_wpd_30by23_a.
question(pacc_wpd_30by4_n of pacc_wpd_30by4) =
  get_pacc_wpd_30by4_n.
question(pacc_wpd_30by4_a of pacc_wpd_30by4) =
  get_pacc_wpd_30by4_a.
question(pacc_wpd_30by5_n of pacc_wpd_30by5) =
  get_pacc_wpd_30by5_n.
question(pacc_wpd_30by5_a of pacc_wpd_30by5) =
  get_pacc_wpd_30by5_a.
question(pacc_wpd_30by6_n of pacc_wpd_30by6) =
get_pacc_wpd_30by6_n.
question(pacc_wpd_30by6_a of pacc_wpd_30by6) =
get_pacc_wpd_30by6_a.


question(bk_vals of trw) = get_bk_vals.
question(foreclosure_weight_val of foreclosure) =
get_fclos_vals.
question(judgment_weight_val of judgment) =
     get_judgment_vals.
question(repo_weight_val of repo) = get_repo_vals.

question(charge_off_n of charge_off) =
get_charge_off_n.
question(charge_off_a of charge_off) =
get_charge_off_a.


question(pd_coll_ac_n of pd_coll_ac) =
get_pd_coll_ac_n.
question(pd_coll_ac_a of pd_coll_ac) =
get_pd_coll_ac_a.

question(no_of_inq of inquiry) = check_recent_inq.
question(curr_acct_n of superpayer) = get_curr_acct_n.

question(pdbydlr_n of pdbydlr) = get_pdbydlr_n.
question(pdbydlr_a of pdbydlr) = get_pdbydlr_a.

set(pacc_wdel_info of trw_info of customer_profile_info
     of
    perserec) = pacc_wdel_info.
set(pacc_wdel60_info of pacc_wdel_info of trw_info of
    customer_profile_info of perserec) =
    pacc_wdel60_info.
set(pacc_wdel90_info of pacc_wdel_info of trw_info of
  customer_profile_info of perserec) =
    pacc_wdel90_info.
set(pacc_wdel120_info of pacc_wdel_info of trw_info of
customer_profile_info of perserec) = pacc_wdel120_info.
set(pacc_wdel150_info of pacc_wdel_info of trw_info of
customer_profile_info of perserec) = pacc_wdel150_info.
set(pacc_wdel180_info of pacc_wdel_info of trw_info of
customer_profile_info of perserec) = pacc_wdel180_info.

name(pacc_wdel_info of trw) = ignore.
name(pacc_wdel60_info of pacc_wdel) = ignore.
name(pacc_wdel90_info of pacc_wdel) = ignore.
name(pacc_wdel120_info of pacc_wdel) = ignore.
name(pacc_wdel150_info of pacc_wdel) = ignore.
name(pacc_wdel180_info of pacc_wdel) = ignore.


question(pacc_wdel60_n of pacc_wdel60) =
get_pacc_wdel60_n.
question(pacc_wdel60_a of pacc_wdel60) =
get_pacc_wdel60_a.
question(pacc_wdel90_n of pacc_wdel90) =
get_pacc_wdel90_n.
question(pacc_wdel90_a of pacc_wdel90) =
get_pacc_wdel90_a.
question(pacc_wdel120_n of pacc_wdel120) =
get_pacc_wdel120_n.
question(pacc_wdel120_a of pacc_wdel120) =
get_pacc_wdel120_a.
question(pacc_wdel150_n of pacc_wdel150) =
get_pacc_wdel150_n.
question(pacc_wdel150_a of pacc_wdel150) =
get_pacc_wdel150_a.
question(pacc_wdel180_n of pacc_wdel180) =
get_pacc_wdel180_n.
question(pacc_wdel180_a of pacc_wdel180) =
get_pacc_wdel180_a.

set(pd_repo_info of trw_info of customer_profile_info
of
perserec) = pd_repo_info.
name(pd_repo_info of trw) = ignore.
question(pd_repo_n of pd_repo) = get_pd_repo_n.
question(pd_repo_a of pd_repo) = get_pd_repo_a.

set(pd_chg_off_info of trw_info of
customer_profile_info of
perserec) = pd_chg_off_info.
name(pd_chg_off_info of trw) = ignore.
question(pd_chg_off_n of pd_chg_off) =
get_pd_chg_off_n.
question(pd_chg_off_a of pd_chg_off) =
get_pd_chg_off_a.

set(pd_foreclo_info of trw_info of
customer_profile_info of
```

```
perserec) = pd_foreclo_info.
name(pd_foreclo_info of trw) = ignore.
question(pd_foreclo_n of pd_foreclo) =
get_pd_foreclo_n.
question(pd_foreclo_a of pd_foreclo) =
get_pd_foreclo_a.

set(bkliqreo_info of trw_info of customer_profile_info
of
perserec) = bkliqreo_info.
name(bkliqreo_info of trw) = ignore.
question(bkliqreo_n of bkliqreo) = get_bkliqreo_n.
question(bkliqreo_a of bkliqreo) = get_bkliqreo_a.


set(settled_info of trw_info of customer_profile_info
of
perserec) = settled_info.
name(settled_info of trw) = ignore.
question(settled_n of settled) = get_settled_n.
question(settled_a of settled) = get_settled_a.


set(bk_adj_pln_info of trw_info of
customer_profile_info of
perserec) = bk_adj_pln_info.
name(bk_adj_pln_info of trw) = ignore.
question(bk_adj_pln_n of bk_adj_pln) =
get_bk_adj_pln_n.
question(bk_adj_pln_a of bk_adj_pln) =
get_bk_adj_pln_a.


set(scnl_nwloc_info of trw_info of
customer_profile_info of
perserec) = scnl_nwloc_info.
name(scnl_nwloc_info of trw) = ignore.
question(scnl_nwloc_n of scnl_nwloc) =
get_scnl_nwloc_n.
question(scnl_nwloc_a of scnl_nwloc) =
get_scnl_nwloc_a.


set(co_now_pay_info of trw_info of
customer_profile_info of
perserec) = co_now_pay_info.
name(co_now_pay_info of trw) = ignore.
question(co_now_pay_n of co_now_pay) =
get_co_now_pay_n.
```

```
question(co_now_pay_a of co_now_pay) =
get_co_now_pay_a.


set(fore_proc_info of trw_info of customer_profile_info
of
perserec) = fore_proc_info.
name(fore_proc_info of trw) = ignore.
question(fore_proc_n of fore_proc) = get_fore_proc_n.
question(fore_proc_a of fore_proc) = get_fore_proc_a.


set(gov_claim_info of trw_info of customer_profile_info
of
perserec) = gov_claim_info.
name(gov_claim_info of trw) = ignore.
question(gov_claim_n of gov_claim) = get_gov_claim_n.
question(gov_claim_a of gov_claim) = get_gov_claim_a.


set(close_np_aa_info of trw_info of
customer_profile_info of
perserec) = close_np_aa_info.
name(close_np_aa_info of trw) = ignore.
question(close_np_aa_n of close_np_aa) =
get_close_np_aa_n.
question(close_np_aa_a of close_np_aa) =
get_close_np_aa_a.

set(scnl_info of trw_info of customer_profile_info of
perserec) = scnl_info.
name(scnl_info of trw) = ignore.
question(scnl_n of scnl) = get_scnl_n.
question(scnl_a of scnl) = get_scnl_a.


set(fed_tax_ln_info of trw_info of
customer_profile_info of
perserec) = fed_tax_ln_info.
name(fed_tax_ln_info of trw) = ignore.
question(fed_tax_ln_n of fed_tax_ln) =
get_fed_tax_ln_n.
question(fed_tax_ln_a of fed_tax_ln) =
get_fed_tax_ln_a.


set(fed_tax_rel_info of trw_info of
customer_profile_info of
perserec) = fed_tax_rel_info.
```

```
name(fed_tax_rel_info of trw) = ignore.
question(fed_tax_rel_n of fed_tax_rel) =
get_fed_tax_rel_n.
question(fed_tax_rel_a of fed_tax_rel) =
get_fed_tax_rel_a.
set(judgmt_sat_info of trw_info of
customer_profile_info of
perserec) = judgmt_sat_info.
name(judgmt_sat_info of trw) = ignore.
question(judgmt_sat_n of judgmt_sat) =
get_judgmt_sat_n.
question(judgmt_sat_a of judgmt_sat) =
get_judgmt_sat_a.


set(judg_vacat_info of trw_info of
customer_profile_info of
perserec) = judg_vacat_info.
name(judg_vacat_info of trw) = ignore.
question(judg_vacat_n of judg_vacat) =
get_judg_vacat_n.
question(judg_vacat_a of judg_vacat) =
get_judg_vacat_a.


set(mech_lien_info of trw_info of customer_profile_info
of
perserec) = mech_lien_info.
name(mech_lien_info of trw) = ignore.
question(mech_lien_n of mech_lien) = get_mech_lien_n.
question(mech_lien_a of mech_lien) = get_mech_lien_a.


set(mech_rele_info of trw_info of customer_profile_info
of
perserec) = mech_rele_info.
name(mech_rele_info of trw) = ignore.
question(mech_rele_n of mech_rele) = get_mech_rele_n.
question(mech_rele_a of mech_rele) = get_mech_rele_a.


set(mn_mtg_fil_info of trw_info of
customer_profile_info of
perserec) = mn_mtg_fil_info.
name(mn_mtg_fil_info of trw) = ignore.
question(mn_mtg_fil_n of mn_mtg_fil) =
get_mn_mtg_fil_n.
question(mn_mtg_fil_a of mn_mtg_fil) =
get_mn_mtg_fil_a.
```

```
set(nt_respon_info of trw_info of customer_profile_info
of
perserec) = nt_respon_info.
name(nt_respon_info of trw) = ignore.
question(nt_respon_n of nt_respon) = get_nt_respon_n.
question(nt_respon_a of nt_respon) = get_nt_respon_a.


set(stat_tx_ln_info of trw_info of
customer_profile_info of
perserec) = stat_tx_ln_info.
name(stat_tx_ln_info of trw) = ignore.
question(stat_tx_ln_n of stat_tx_ln) =
get_stat_tx_ln_n.
question(stat_tx_ln_a of stat_tx_ln) =
get_stat_tx_ln_a.

set(sta_tx_rel_info of trw_info of
customer_profile_info of
perserec) = sta_tx_rel_info.
name(sta_tx_rel_info of trw) = ignore.
question(sta_tx_rel_n of sta_tx_rel) =
get_sta_tx_rel_n.
question(sta_tx_rel_a of sta_tx_rel) =
get_sta_tx_rel_a.


set(suit_dismd_info of trw_info of
customer_profile_info of
perserec) = suit_dismd_info.
name(suit_dismd_info of trw) = ignore.
question(suit_dismd_n of suit_dismd) =
get_suit_dismd_n.
question(suit_dismd_a of suit_dismd) =
get_suit_dismd_a.


set(wage_asign_info of trw_info of
customer_profile_info of
perserec) = wage_asign_info.
name(wage_asign_info of trw) = ignore.
question(wage_asign_n of wage_asign) =
get_wage_asign_n.
question(wage_asign_a of wage_asign) =
get_wage_asign_a.
```

```
set(wa_release_info of trw_info of
customer_profile_info of
perserec) = wa_release_info.
name(wa_release_info of trw) = ignore.
question(wa_release_n of wa_release) =
get_wa_release_n.
question(wa_release_a of wa_release) =
get_wa_release_a.

set(refinanced_info of trw_info of
customer_profile_info of
perserec) = refinanced_info.
name(refinanced_info of trw) = ignore.
question(refinanced_n of refinanced) =
get_refinanced_n.
question(refinanced_a of refinanced) =
get_refinanced_a.


set(cr_cd_lost_info of trw_info of
customer_profile_info of
perserec) = cr_cd_lost_info.
name(cr_cd_lost_info of trw) = ignore.
question(cr_cd_lost_n of cr_cd_lost) =
get_cr_cd_lost_n.
question(cr_cd_lost_a of cr_cd_lost) =
get_cr_cd_lost_a.


set(clos_inac_info of trw_info of customer_profile_info
of
perserec) = clos_inac_info.
name(clos_inac_info of trw) = ignore.
question(clos_inac_n of clos_inac) = get_clos_inac_n.
question(clos_inac_a of clos_inac) = get_clos_inac_a.


set(transfered_info of trw_info of
customer_profile_info of
perserec) = transfered_info.
name(transfered_info of trw) = ignore.
question(transfered_n of transfered) =
get_transfered_n.
question(transfered_a of transfered) =
get_transfered_a.


set(too_new_rt_info of trw_info of
customer_profile_info of
perserec) = too_new_rt_info.
name(too_new_rt_info of trw) = ignore.
question(too_new_rt_n of too_new_rt) =
get_too_new_rt_n.
question(too_new_rt_a of too_new_rt) =
get_too_new_rt_a.


set(paid_satis_info of trw_info of
customer_profile_info of
perserec) = paid_satis_info.
name(paid_satis_info of trw) = ignore.
question(paid_satis_n of paid_satis) =
get_paid_satis_n.
question(paid_satis_a of paid_satis) =
get_paid_satis_a.


set(paid_acct_info of trw_info of customer_profile_info
of
perserec) = paid_acct_info.
name(paid_acct_info of trw) = ignore.
question(paid_acct_n of paid_acct) = get_paid_acct_n.
question(paid_acct_a of paid_acct) = get_paid_acct_a.


set(deceased_info of trw_info of customer_profile_info
of
perserec) = deceased_info.
name(deceased_info of trw) = ignore.
question(deceased_n of deceased) = get_deceased_n.
question(deceased_a of deceased) = get_deceased_a.

set(cr_ln_clos_info of trw_info of
customer_profile_info of
perserec) = cr_ln_clos_info.
name(cr_ln_clos_info of trw) = ignore.
question(cr_ln_clos_n of cr_ln_clos) =
get_cr_ln_clos_n.
question(cr_ln_clos_a of cr_ln_clos) =
get_cr_ln_clos_a.


set(redmd_repo_info of trw_info of
customer_profile_info of
perserec) = redmd_repo_info.
name(redmd_repo_info of trw) = ignore.
question(redmd_repo_n of redmd_repo) =
get_redmd_repo_n.
```

```
question(redmd_repo_a of redmd_repo) =
get_redmd_repo_a.


set(cur_was_col_info of trw_info of
customer_profile_info of
perserec) = cur_was_col_info.
name(cur_was_col_info of trw) = ignore.
question(cur_was_col_n of cur_was_col) =
get_cur_was_col_n.
question(cur_was_col_a of cur_was_col) =
get_cur_was_col_a.
set(cr_ln_rnst_info of trw_info of
customer_profile_info of
perserec) = cr_ln_rnst_info.
name(cr_ln_rnst_info of trw) = ignore.
question(cr_ln_rnst_n of cr_ln_rnst) =
get_cr_ln_rnst_n.
question(cr_ln_rnst_a of cr_ln_rnst) =
get_cr_ln_rnst_a.


set(cur_was_for_info of trw_info of
customer_profile_info of
perserec) = cur_was_for_info.
name(cur_was_for_info of trw) = ignore.
question(cur_was_for_n of cur_was_for) =
get_cur_was_for_n.
question(cur_was_for_a of cur_was_for) =
get_cur_was_for_a.
set(pd_not_aa_info of trw_info of customer_profile_info
of
perserec) = pd_not_aa_info.
name(pd_not_aa_info of trw) = ignore.
question(pd_not_aa_n of pd_not_aa) = get_pd_not_aa_n.
question(pd_not_aa_a of pd_not_aa) = get_pd_not_aa_a.


set(city_tx_ln_info of trw_info of
customer_profile_info of
perserec) = city_tx_ln_info.
name(city_tx_ln_info of trw) = ignore.
question(city_tx_ln_n of city_tx_ln) =
get_city_tx_ln_n.
question(city_tx_ln_a of city_tx_ln) =
get_city_tx_ln_a.
```

```
set(city_tx_rel_info of trw_info of
customer_profile_info of
perserec) = city_tx_rel_info.
name(city_tx_rel_info of trw) = ignore.
question(city_tx_rel_n of city_tx_rel) =
get_city_tx_rel_n.
question(city_tx_rel_a of city_tx_rel) =
get_city_tx_rel_a.
set(consel_ser_info of trw_info of
customer_profile_info of
perserec) = consel_ser_info.
name(consel_ser_info of trw) = ignore.
question(consel_ser_n of consel_ser) =
get_consel_ser_n.
question(consel_ser_a of consel_ser) =
get_consel_ser_a.


set(co_tax_ln_info of trw_info of customer_profile_info
of
perserec) = co_tax_ln_info.
name(co_tax_ln_info of trw) = ignore.
question(co_tax_ln_n of co_tax_ln) = get_co_tax_ln_n.
question(co_tax_ln_a of co_tax_ln) = get_co_tax_ln_a.


set(co_tax_rel_info of trw_info of
customer_profile_info of
perserec) = co_tax_rel_info.
name(co_tax_rel_info of trw) = ignore.
question(co_tax_rel_n of co_tax_rel) =
get_co_tax_rel_n.
question(co_tax_rel_a of co_tax_rel) =
get_co_tax_rel_a.

synonym(concluded)= $ the conclusion $.
synonym(conclusion)= $ the current case $.
synonym(weight_pt)=$ the current case $.
synonym(overall_weight_val)=$overall calculated
weight$.
synonym(weight_pt_info)=$ current case's total
weights$.
synonym(weight_pt_val) = $above weight$.
synonym(cut_off)=$the range 0-200 that we use$.
synonym(cut_off_info)=$determined cut off values$.
synonym(potential_bankrupt)=$potential bankrupt$.
synonym(pb_cut_off_val)=$potential bankrupt cut off
value$.
```

```
synonym(very_critical_cut_off_val)=$very critical cut
off value$.

synonym(critical_cut_off_val)=$critical cut off value$.

synonym(serious_cut_off_val)=$serious cut off value$.
synonym(very_poor_cut_off_val)=$very poor cut off
value$.
synonym(poor_cut_off_val)=$poor cut off value$.
synonym(satisfactory_cut_off_val)=$satisfactory cut off
value$.
synonym(normal_cut_off_val)=$normal cut off value$.



/* Synonyms for easier user recognition */

synonym(conclusion_info)=$current case$.
synonym(too_many_accts)=$Too many accounts$.
synonym(curr_acct_n)=$number of current accounts$.
synonym(superpayer_info)=$Superpayer$.
synonym(alarmingly_high)=$alarmingly high$.
synonym(extremely_low)=$extremely low$.
synonym(extremely_low_cut_off_val)=$the extremely low
    cut off value$.
synonym(very_low)=$very low$.
synonym(very_low_cut_off_val)=$very low cut off value$.
synonym(low_cut_off_val)=$low cut off value$.
synonym(very_poor)=$very poor$.
synonym(moderately_high_cut_off_val)=$moderately high
    cut off value$.
synonym(high_cut_off_val)=$high cut off value$.
synonym(very_high_cut_off_val)=$very high cut off
value$.
synonym(very_critical)=$very critical$.
synonym(extremely_high)=$extremely high$.
synonym(extremely_high_cut_off_val)=$extremely high cut
    off value$.
synonym(public_record_val)=$public record val$.
synonym(private_record_val)=$private record value$.
synonym(weighted_decision)=$weighted decision$.
synonym(set_of_bkrpts_and_liens_weight_val)=$the
    calculated weights for all bankruptcies and liens$.
synonym(set_of_misc_public_items_weight_val)=$the
    calculated weights for all miscellaneous public
items$.
synonym(set_of_current_and_paid_accts_weight_val)=$the
    calculated weights for all current and paid
accounts$.
```

```
synonym(set_of_delinq_accts_weight_val)=$the calculated
    weights for all delinquent accounts$.
synonym(set_of_misc_private_items_weight_val)=$the
    calculated weights for all miscellaneous private
    items$.
synonym(bk_adj_pln_weight_val)=$the calculated weight
    for debt included or discharged through bankruptcy
    chapter 13$.
synonym(bk_adj_pln_info)=$the current case$.
synonym(bkliqreo_weight_val)=$the calculated weight for
    debt included or discharged through bankruptcy
    chapter 7 or 11$.
synonym(bkliqreo_info)=$the current case$.
synonym(bk_vals)=$the calculated weight for bankruptcy
    filed, dismissed or completed in chapters 7, 11 or
    13$.
synonym(trw_info)=$the current case$.
synonym(mech_lien_weight_val)=$the calculated weight
    for mechanics lien$.
synonym(mech_lien_info)=$the current case$.
synonym(mech_rele_weight_val)=$the calculated weight
    for mechanics lien released$.
synonym(mech_rele_info)=$the current case$.
synonym(fed_tax_ln_weight_val)=$the calculated weight
    for federal tax lien$.
synonym(fed_tax_ln_info)=$the current case$.
synonym(fed_tax_rel_weight_val)=$the calculated weight
    for federal tax lien release$.
synonym(fed_tax_rel_info)=$the current case$.
synonym(stat_tx_ln_weight_val)=$the calculated weight
    for state tax lien$.
synonym(stat_tx_ln_info)=$the current case$.
synonym(sta_tx_rel_weight_val)=$the calculated weight
    for state tax lien release$.
synonym(sta_tx_rel_info)=$the current case$.
synonym(city_tx_ln_weight_val)=$the calculated weight
    for city tax lien$.
synonym(city_tx_ln_info)=$the current case$.
synonym(city_tx_rel_weight_val)=$the calculated weight
    for city tax lien release$.
synonym(city_tx_rel_info)=$the current case$.
synonym(co_tax_ln_weight_val)=$the calculated weight
    for county tax lien$.
synonym(co_tax_ln_info)=$the current case$.
synonym(co_tax_rel_weight_val)=$the calculated weight
    for county tax lien release$.
synonym(co_tax_rel_info)=$the current case$.
synonym(judgmt_sat_weight_val)=$the calculated weight
    for satisfied judgement$.
```

```
synonym(judgmt_sat_info)=$the current case$.
synonym(judg_vacat_weight_val)=$the calculated weight
    for vacated judgement$.
synonym(judg_vacat_info)=$the current case$.
synonym(judgment_weight_val)=$the calculated weight for
    judgement$.
synonym(judgment_info)=$the current case$.
synonym(wage_asign_weight_val)=$the calculated weight
    for wage assignment$.
synonym(wage_asign_info)=$the current case$.
synonym(wa_release_weight_val)=$the calculated weight
    for wage assignment released$.
synonym(wa_release_info)=$the current case$.
synonym(suit_weight_val)=$the calculated weight for
    suit$.
synonym(suit_info)=$the current case$.
synonym(suit_dismd_weight_val)=$the calculated weight
    for suit dismissed$.
synonym(mn_mtg_fil_weight_val)=$the calculated weight
    for filed manual mortgage report$.
synonym(mn_mtg_fil_info)=$the current case$.
synonym(nt_respon_weight_val)=$the calculated weight
    for not responsible notice$.
synonym(nt_respon_info)=$the current case$.
synonym(consel_ser_weight_val)=$the calculated weight
    for debt counseling service$.
synonym(consel_ser_info)=$the current case$.
synonym(curr_acct_n_weight_val)=$the calculated weight
    for number of current accounts$.
synonym(superpayer_info)=$the current case$.
synonym(cur_was_col_weight_val)=$the calculated weight
    for a current account that was a collection
    account$.
synonym(cur_was_col_info)=$the current case$.
synonym(cur_was_for_weight_val)=$the calculated weight
    for a current account that was a foreclosure$.
synonym(cur_was_for_info)=$the current case$.
synonym(cur_was_weight_val)=$the calculated weight for
    a current account that was delinquent$.
synonym(cur_was_info)=$the current case$.
synonym(cwpd_weight_val)=$the calculated weight for a
    current account that was past due$.
synonym(curwaspd_info)=$the current case$.
synonym(paid_satis_weight_val)=$the calculated weight
    for a closed account that was paid satisfactorily$.
synonym(paid_satis_info)=$the current case$.
synonym(paid_acct_weight_val)=$the calculated weight
    for an account that is closed, has zero balance or
    is not rated$.
```

```
synonym(paid_acct_info)=$the current case$.
synonym(pd_not_aa_weight_val)=$the calculated weight
    for a paid account with some payments not made as
    agreed$.
synonym(pd_not_aa_info)=$the current case$.
synonym(pacc_wpd_weight_val)=$the calculated weight for
    a paid account that was past due$.
synonym(pacc_wpd_info)=$the current case$.
synonym(pacc_wdel_weight_val)=$the calculated weight
    for a paid account that was delinquent$.
synonym(pacc_wdel_info)=$the current case$.
synonym(pd_chg_off_weight_val)=$the calculated weight
    for a paid account that was a charge off$.
synonym(pd_chg_off_info)=$the current case$.
synonym(pd_repo_weight_val)=$the calculated weight for
    a paid account that was a repossession$.
synonym(pd_repo_info)=$the current case$.
synonym(pd_coll_ac_weight_val)=$the calculated weight
    for a paid account that was a collection, insurance
    or education claim$.
synonym(pd_coll_ac_info)=$the current case$.
synonym(notpdaa_weight_val)=$the calculated weight for
    an account not being paid as agreed$.
synonym(notpdaa_info)=$the current case$.
synonym(pd_foreclo_weight_val)=$the calculated weight
    for a paid account that was a foreclosure$.
synonym(pd_foreclo_info)=$the current case$.
synonym(delinq_weight_val)=$the calculated weight for a
    delinquent account$.
synonym(delinq_info)=$the current case$.
synonym(accpd_weight_val)=$the calculated weight for an
    account past due$.
synonym(accpd_info)=$the current case$.
synonym(volnrepo_weight_val)=$the calculated weight for
    voluntary repossession$.
synonym(volnrepo_info)=$the current case$.
synonym(foreclosure_weight_val)=$the calculated weight
    for a foreclosure$.
synonym(foreclosure_info)=$the current case$.
synonym(repo_weight_val)=$the calculated weight for a
    repossession$.
synonym(repo_info)=$the current case$.
synonym(pdbydlr_weight_val)=$the calculated weight for
    an account that was paid by dealer$.
synonym(pdbydlr_info)=$the current case$.
synonym(coll_acct_weight_val)=$the calculated weight
    for a collection account$.
synonym(coll_acct_info)=$the current case$.
synonym(charge_off_weight_val)=$the calculated weight
    for a charge off$.
synonym(charge_off_info)=$the current case$.
synonym(close_np_aa_weight_val)=$the calculated weight
    for credit line closed account not paid as agreed$.
synonym(close_np_aa_info)=$the current case$.
synonym(scnl_weight_val)=$the calculated weight for
    unlocated consumer$.
synonym(scnl_info)=$the current case$.
synonym(fore_proc_weight_val)=$the calculated weight
    for foreclosure proceeding$.
synonym(fore_proc_info)=$the current case$.
synonym(insclaim_weight_val)=$the calculated weight for
    claim filed against insured portion of balance$.
synonym(insclaim_info)=$the current case$.
synonym(gov_claim_weight_val)=$the calculated weight
    for government claim$.
synonym(gov_claim_info)=$the current case$.
synonym(settled_weight_val)=$the calculated weight for
    a settled account$.
synonym(settled_info)=$the current case$.
synonym(scnl_nwloc_weight_val)=$the calculated weight
    for a located consumer previously unlocated$.
synonym(scnl_nwloc_info)=$the current case$.
synonym(co_now_pay_weight_val)=$the calculated weight
    for a paying account that was a charge off$.
synonym(co_now_pay_info)=$the current case$.
synonym(refinanced_weight_val)=$the calculated weight
    for a refinanced account$.
synonym(refinanced_info)=$the current case$.
synonym(cr_cd_lost_weight_val)=$the calculated weight
    for credit card lost or stolen$.
synonym(cr_cd_lost_info)=$the current case$.
synonym(clos_inac_weight_val)=$the calculated weight
    for closed inactive account$.
synonym(clos_inac_info)=$the current case$.
synonym(transfered_weight_val)=$the calculated weight
    for transferred account$.
synonym(transfered_info)=$the current case$.
synonym(too_new_rt_weight_val)=$the calculated weight
    for an account too new to rate$.
synonym(too_new_rt_info)=$the current case$.
synonym(deceased_weight_val)=$the calculated weight for
    deceased$.
synonym(deceased_info)=$the current case$.
synonym(cr_ln_clos_weight_val)=$the calculated weight
    for credit line closed$.
synonym(cr_ln_clos_info)=$the current case$.
synonym(redmd_repo_weight_val)=$the calculated weight
    for redeemed repossession$.
synonym(redmd_repo_info)=$the current case$.
synonym(cr_ln_rnst_weight_val)=$the calculated weight
for credit line reinstated$.
synonym(cr_ln_rnst_info)=$the current case$.
synonym(inq_weight_val)=$the calculated weight for
    inquiry$.
synonym(inquiry_info)=$the current case$.
synonym(superpayer_weight_val)=$the calculated weight
    for superpayer$.
synonym(superpayer_info)=$the current case$.
synonym(bkliqreo_n)=$the number of occurrences of debt
    included or discharged through bankruptcy chapter 7
    or 11 $.
synonym(settled_n)=$the number of occurrences of
    settled accounts$.
synonym(bk_adj_pln_n)=$the number of occurrences of
    debt included or discharged through bankruptcy
    chapter 13$.
synonym(scnl_nwloc_n)=$the number of occurrences of a
    located consumer previously unlocated$.
synonym(co_now_pay_n)=$the number of occurrences of
charge offs now being paid$.
synonym(fore_proc_n)=$the number of occurrences of
    foreclosure proceedings$.
synonym(gov_claim_n)=$the number of occurrences of
    government claims$.
synonym(close_np_aa_n)=$the number of occurrences of
    credit line closed account not paid as agreed$.
synonym(scnl_n)=$the number of occurrences of unlocated
    consumer$.
synonym(fed_tax_ln_n)=$the number of occurrences of
    federal tax liens$.
synonym(fed_tax_rel_n)=$the number of occurrences of
    federal tax liens released$.
synonym(judgmt_sat_n)=$the number of occurrences of
    satisfactory judgements$.
synonym(judg_vacat_n)=$the number of occurrences of
vacated judgements$.
synonym(mech_lien_n)=$the number of occurrences of
    mechanics liens$.
synonym(mech_rele_n)=$the number of occurrences of
    mechanics liens released$.
synonym(mn_mtg_fil_n)=$the number of occurrences of
    filed manual mortgage reports$.
synonym(nt_respon_n)=$the number of occurrences of not
    responsible notices$.
```

```
synonym(stat_tx_ln_n)=$the number of occurrences of
    state tax liens$.
synonym(sta_tx_rel_n)=$the number of occurrences of
    state tax liens released$.
synonym(suit_dismd_n)=$the number of occurrences of
    suits$.
synonym(wage_asign_n)=$the number of occurrences of
    wage assignments$.
synonym(wa_release_n)=$the number of occurrences of
    wage assignments released$.
synonym(refinanced_n)=$the number of occurrences of
    refinanced$.
synonym(cr_cd_lost_n)=$the number of occurrences of
    credit card lost$.
synonym(clos_inac_n)=$the number of occurrences of
    closed inactive account$.
synonym(transfered_n)=$the number of occurrences of
    transfers$.
synonym(too_new_rt_n)=$the number of occurrences of too
    new to rate$.
synonym(paid_satis_n)=$the number of occurrences of
    paid satisfactorily$.
synonym(paid_acct_n)=$the number of occurrences of paid
    account$.
synonym(deceased_n)=$the number of occurrences of
    deceased$.
synonym(cr_ln_clos_n)=$the number of occurrences of
    credit line closed$.
synonym(redmd_repo_n)=$the number of occurrences of
    redeemed repossessions$.
synonym(cur_was_col_n)=$the number of occurrences of
    current accounts that were collection accounts$.
synonym(cr_ln_rnst_n)=$the number of occurrences of
    credit line reinstated$.
synonym(cur_was_for_n)=$the number of occurrences of
    current accounts that were foreclosures$.
synonym(pd_not_aa_n)=$the number of occurrences of paid
    accounts with some payments not made as agreed$.
synonym(city_tx_ln_n)=$the number of occurrences of
    city tax lien$.
synonym(city_tx_rel_n)=$the number of occurrences of
    city tax lien released$.
synonym(consel_ser_n)=$the number of occurrences of
    debt counseling service$.
synonym(co_tax_ln_n)=$the number of occurrences of
    county tax lien$.
synonym(co_tax_rel_n)=$the number of occurrences of
    county tax lien released$.
```

```
synonym(pd_foreclo_n)=$the number of occurrences of
    paid accounts that were foreclosures$.
synonym(pd_chg_off_n)=$the number of occurrences of
    paid accounts that were charge offs$.
synonym(pd_repo_n)=$the number of occurrences of paid
    accounts that were repossessions$.
synonym(pd_coll_ac_n)=$the number of occurrences of
    paid accounts that were collections, or insurance
    or education claims$.
synonym(pacc_wdel60_n)=$the number of occurrences of
    paid accounts that were delinquent 60 days$.
synonym(pacc_wdel90_n)=$the number of occurrences of
    paid accounts that were delinquent 90 days$.
synonym(pacc_wdel120_n)=$the number of occurrences of
    paid accounts that were delinquent 120 days$.
synonym(pacc_wdel150_n)=$the number of occurrences of
    paid accounts that were delinquent 150 days$.
synonym(pacc_wdel180_n)=$the number of occurrences of
    paid accounts that were delinquent 180 days$.
synonym(suit_n)=$the number of occurrences of suits$.
synonym(pacc_wpd_30_n)=$the number of occurrences of
    paid accounts that were past due 30 days$.
synonym(pacc_wpd_30by23_n)=$the number of occurrences
    of paid accounts that were past due 30 days 2 or 3
    times$.
synonym(pacc_wpd_30by4_n)=$the number of occurrences of
    paid accounts that were past due 30 days 4 times$.
synonym(pacc_wpd_30by5_n)=$the number of occurrences of
    paid accounts that were past due 30 days 5 times$.
synonym(pacc_wpd_30by6_n)=$the number of occurrences of
    paid accounts that were past due 30 days 6 times$.
synonym(accpd_30_n)=$the number of occurrences of an
    account past due 30 days$.
synonym(accpd_30by2_n)=$the number of occurrences of an
    account past due 30 days 2 times$.
synonym(accpd_30by3_n)=$the number of occurrences of an
    account past due 30 days 3 times$.
synonym(accpd_30by4_n)=$the number of occurrences of an
    account past due 30 days 4 times$.
synonym(accpd_30by5_n)=$the number of occurrences of an
    account past due 30 days 5 times$.
synonym(accpd_30by6_n)=$the number of occurrences of an
    account past due 30 days 6 times$.
synonym(cwpd_n)=$the number of occurrences of a current
    account that was past due$.
synonym(cwpd_30_n)=$the number of occurrences of a
    current account that was past due 30 days$.
synonym(cwpd_30by2_n)=$the number of occurrences of a
```

```
    current account that was past due 30 days 2
times$.
synonym(cwpd_30by3_n)=$the number of occurrences of a
    current account that was past due 30 days 3 times$.
synonym(cwpd_30by4_n)=$the number of occurrences of a
    current account that was past due 30 days 4 times$.
synonym(cwpd_30by5_n)=$the number of occurrences of a
    current account that was past due 30 days 5 times$.
synonym(cwpd_30by6_n)=$the number of occurrences of a
    current account that was past due 30 days 6 times$.
synonym(d_60_n)=$the number of occurrences of an
    account delinquent 60 days$.
synonym(d_90_n)=$the number of occurrences of an
    account delinquent 90 days$.
synonym(d_120_n)=$the number of occurrences of an
    account delinquent 120 days$.
synonym(d_150_n)=$the number of occurrences of an
    account delinquent 150 days$.
synonym(d_180_n)=$the number of occurrences of an
    account delinquent 180 days$.
synonym(c_w_60_n)=$the number of occurrences of a
    current account that was delinquent 60 days$.
synonym(c_w_90_n)=$the number of occurrences of a
    current account that was delinquent 90 days$.
synonym(c_w_120_n)=$the number of occurrences of a
    current account that was delinquent 120 days$.
synonym(c_w_150_n)=$the number of occurrences of a
    current account that was delinquent 150 days$.
synonym(c_w_180_n)=$the number of occurrences of a
    current account that was delinquent 180 days$.
synonym(pdbydlr_n)=$the number of occurrences of
    accounts paid by dealer$.
synonym(coll_acct_n)=$the number of occurrences of
    collection account$.
synonym(charge_off_n)=$the number of occurrences of
    charge offs$.
synonym(curr_acct_n)=$the number of occurrences of
current accounts$.
synonym(insclaim_n)=$the number of occurrences of a
    claim filed against insured portion of balance$.
synonym(notpdaa_n)=$the number of occurrences of an
    account not being paid as agreed$.
synonym(volnrepo_n)=$the number of occurrences of
    voluntary repossession$.


         /* ----------- RULES ------------ */

the concluded of conclusion_info is too_many_accts
```

```
   if
the curr_acct_n of superpayer_info is N and
N > 20.

the concluded of conclusion_info is potential_bankrupt
     if
the weight_pt_val of weight_pt_info is alarmingly_high.


the weight_pt_val of weight_pt_info is alarmingly_high
     if
the overall_weight_val of conclusion_info is Y and
the pb_cut_off_val of cut_off_info is Z and
Y > Z.

the concluded of conclusion_info is superpayer
     if
the monthly_payment of the superpayer_info is E and
the monthly_income of the personal_info is I and
E > I.

the concluded of conclusion_info is normal
     if
the weight_pt_val of weight_pt_info is extremely_low.

the weight_pt_val of weight_pt_info is extremely_low
     if
the overall_weight_val of conclusion_info is Y and
the extremely_low_cut_off_val of cut_off_info is Z and
Y =< Z.

the concluded of conclusion_info is satisfactory
     if
the weight_pt_val of weight_pt_info is very_low.

the weight_pt_val of weight_pt_info is very_low
     if
the overall_weight_val of conclusion_info is Y and
the very_low_cut_off_val of cut_off_info is Z and
Y =< Z.

the concluded of conclusion_info is poor
     if
the weight_pt_val of weight_pt_info is low.

the weight_pt_val of weight_pt_info is low
     if
the overall_weight_val of conclusion_info is Y and
the low_cut_off_val of cut_off_info is Z and
```

```
Y =< Z.

the concluded of conclusion_info is very_poor
     if
the weight_pt_val of weight_pt_info is moderately_high.


the weight_pt_val of weight_pt_info is moderately_high
     if
the overall_weight_val of conclusion_info is Y and
the moderately_high_cut_off_val of cut_off_info is Z
and
Y =< Z.

the concluded of conclusion_info is serious
     if
the weight_pt_val of weight_pt_info is high.

the weight_pt_val of weight_pt_info is high
     if
the overall_weight_val of conclusion_info is Y and
the high_cut_off_val of cut_off_info is Z and
Y =< Z.

the concluded of conclusion_info is critical
     if
the weight_pt_val of weight_pt_info is very_high.

the weight_pt_val of weight_pt_info is very_high
     if
the overall_weight_val of conclusion_info is Y and
the very_high_cut_off_val of cut_off_info is Z and
Y =< Z.

the concluded of conclusion_info is very_critical
     if
the weight_pt_val of weight_pt_info is extremely_high.

the weight_pt_val of weight_pt_info is extremely_high
     if
the overall_weight_val of conclusion_info is Y and
the extremely_high_cut_off_val of cut_off_info is Z and

Y > Z.


the overall_weight_val of conclusion_info is Y
     if
the public_record_val of conclusion_info is Y_1 and
the private_record_val of conclusion_info is Y_2 and
Y = Y_1+Y_2.

the concluded of conclusion_info is X
     if
the overall_weight_val of conclusion_info is Y and

weighted_decision(Y,X).


the public_record_val of conclusion_info is Z
     if
the set_of_bkrpts_and_liens_weight_val of
conclusion_info is Z_1 and
the set_of_misc_public_items_weight_val of
conclusion_info is Z_2
and Z = Z_1+Z_2.

the private_record_val of conclusion_info is Z
     if
the set_of_current_and_paid_accts_weight_val of
conclusion_info is Z_3 and
the set_of_delinq_accts_weight_val of conclusion_info
is Z_4 and
the set_of_misc_private_items_weight_val of
conclusion_info is Z_5 and
Z = Z_3+Z_4+Z_5.


the set_of_bkrpts_and_liens_weight_val of
conclusion_info is Z_1
     if
the bk_adj_pln_weight_val of bk_adj_pln_info is X_1 and
the bkliqreo_weight_val of bkliqreo_info is X_2 and
the bk_vals of trw_info is X_3 and
the mech_lien_weight_val of mech_lien_info is X_4 and
the mech_rele_weight_val of mech_rele_info is X_5 and
the fed_tax_ln_weight_val of fed_tax_ln_info is X_6 and
the fed_tax_rel_weight_val of fed_tax_rel_info is X_7
     and
the stat_tx_ln_weight_val of stat_tx_ln_info is X_8 and
the sta_tx_rel_weight_val of sta_tx_rel_info is X_9 and
the city_tx_ln_weight_val of city_tx_ln_info is X_10
     and
the city_tx_rel_weight_val of city_tx_rel_info is X_11
and
the co_tax_ln_weight_val of co_tax_ln_info is X_12 and
the co_tax_rel_weight_val of co_tax_rel_info is X_13
     and
```

```
Z_1 = X_1+X_2+X_3+X_4+X_5+X_6+X_7+X_8+X_9+X_10
     +X_11+X_12+X_13.

the set_of_misc_public_items_weight_val of
conclusion_info is Z_2
     if
the judgmt_sat_weight_val of judgmt_sat_info is X_20
    and
the judg_vacat_weight_val of judg_vacat_info is X_21
    and
the judgment_weight_val of judgment_info is X_22 and
the wage_asign_weight_val of wage_asign_info is X_23
    and
the wa_release_weight_val of wa_release_info is X_24
    and
the suit_dismd_weight_val of suit_dismd_info is X_25
    and
the suit_weight_val of suit_info is X_26 and
the mn_mtg_fil_weight_val of mn_mtg_fil_info is X_27
    and
the nt_respon_weight_val of nt_respon_info is X_28 and
the consel_ser_weight_val of consel_ser_info is X_29
    and
Z_2 = X_20+X_21+X_22+X_23+X_24+X_25+X_26+X_27
      +X_28+X_29.


the set_of_current_and_paid_accts_weight_val of
conclusion_info is Z_3
     if
the curr_acct_n_weight_val of superpayer_info is X_30
and
the cur_was_col_weight_val of cur_was_col_info is X_31
    and
the cur_was_for_weight_val of cur_was_for_info is X_32
    and
the cur_was_weight_val of cur_was_info is X_33 and
the cwpd_weight_val of curwaspd_info is X_34 and
the paid_satis_weight_val of paid_satis_info is X_35
    and
the paid_acct_weight_val of paid_acct_info is X_36 and
the pd_not_aa_weight_val of pd_not_aa_info is X_37 and
the pacc_wpd_weight_val of pacc_wpd_info is X_38 and
the pacc_wdel_weight_val of pacc_wdel_info is X_39 and
the pd_chg_off_weight_val of pd_chg_off_info is X_40
    and
the pd_repo_weight_val of pd_repo_info is X_41 and
the pd_coll_ac_weight_val of pd_coll_ac_info is X_42
    and
the notpdaa_weight_val of notpdaa_info is X_43 and
the pd_foreclo_weight_val of pd_foreclo_info is X_44
    and
Z_3=X_30+X_31+X_32+X_33+X_34+X_35+X_36+X_37+X_38+
    X_39+X_40+X_41+X_42+X_43+X_44.

the set_of_delinq_accts_weight_val of conclusion_info
is Z_4
    if
the delinq_weight_val of delinq_info is X_45 and
the accpd_weight_val of accpd_info is X_46 and
the volnrepo_weight_val of volnrepo_info is X_47 and
the foreclosure_weight_val of foreclosure_info is X_48
    and
the repo_weight_val of repo_info is X_49 and
the pdbydlr_weight_val of pdbydlr_info is X_50 and
the coll_acct_weight_val of coll_acct_info is X_51 and
the charge_off_weight_val of charge_off_info is X_52
    and
the close_np_aa_weight_val of close_np_aa_info is X_53
    and
the scnl_weight_val of scnl_info is X_54 and
the fore_proc_weight_val of fore_proc_info is X_55 and
the insclaim_weight_val of insclaim_info is X_56 and
the gov_claim_weight_val of gov_claim_info is X_57 and
Z_4=X_45+X_46+X_47+X_48+X_49+X_50+X_51+X_52+X_53+
    X_54+X_55+X_56+X_57.

the set_of_misc_private_items_weight_val of
conclusion_info is Z_5
    if
the settled_weight_val of settled_info is X_58 and
the scnl_nwloc_weight_val of scnl_nwloc_info is X_59
    and
the co_now_pay_weight_val of co_now_pay_info is X_60
    and
the refinanced_weight_val of refinanced_info is X_61
    and
the cr_cd_lost_weight_val of cr_cd_lost_info is X_62
    and
the clos_inac_weight_val of clos_inac_info is X_63 and
the transfered_weight_val of transfered_info is X_64
    and
the too_new_rt_weight_val of too_new_rt_info is X_65
    and
the deceased_weight_val of deceased_info is X_66 and
the cr_ln_clos_weight_val of cr_ln_clos_info is X_67
    and
the redmd_repo_weight_val of redmd_repo_info is X_68
    and
the cr_ln_rnst_weight_val of cr_ln_rnst_info is X_69
and
the inq_weight_val of inquiry_info is X_70 and
the superpayer_weight_val of superpayer_info is X_71
    and
Z_5= X_58+X_59+X_60+X_61+X_62+X_63+X_64+X_65+X_66+
     X_67+X_68+X_69+X_70+X_71.


the bkliqreo_weight_val of bkliqreo_info is X
     if
the bkliqreo_n of bkliqreo_info is N_1 and
weight_per_status(bkliqreo,WT_1) and
         X = N_1*WT_1.

the settled_weight_val of settled_info is X
     if
the settled_n of settled_info is N_1 and
weight_per_status(settled,WT_1) and
         X = N_1*WT_1.


the bk_adj_pln_weight_val of bk_adj_pln_info is X
 if
the bk_adj_pln_n of bk_adj_pln_info is N_1 and
weight_per_status(bk_adj_pln,WT_1) and
         X = N_1*WT_1.


the scnl_nwloc_weight_val of scnl_nwloc_info is X
 if
the scnl_nwloc_n of scnl_nwloc_info is N_1 and
weight_per_status(scnl_nwloc,WT_1) and
         X = N_1*WT_1.


the co_now_pay_weight_val of co_now_pay_info is X
 if
the co_now_pay_n of co_now_pay_info is N_1 and
weight_per_status(co_now_pay,WT_1) and
         X = N_1*WT_1.

the fore_proc_weight_val of fore_proc_info is X
      if
the fore_proc_n of fore_proc_info is N_1 and
```

```
weight_per_status(fore_proc,WT_1) and
         X = N_1*WT_1.


the gov_claim_weight_val of gov_claim_info is X
      if
the gov_claim_n of gov_claim_info is N_1 and
weight_per_status(gov_claim,WT_1) and
         X = N_1*WT_1.


the close_np_aa_weight_val of close_np_aa_info is X
 if
the close_np_aa_n of close_np_aa_info is N_1 and
weight_per_status(close_np_aa,WT_1) and
         X = N_1*WT_1.

the scnl_weight_val of scnl_info is X
     if
the scnl_n of scnl_info is N_1 and
weight_per_status(scnl,WT_1) and
         X = N_1*WT_1.


the fed_tax_ln_weight_val of fed_tax_ln_info is X
 if
the fed_tax_ln_n of fed_tax_ln_info is N_1 and
weight_per_status(fed_tax_ln,WT_1) and
         X = N_1*WT_1.


the fed_tax_rel_weight_val of fed_tax_rel_info is X
 if
the fed_tax_rel_n of fed_tax_rel_info is N_1 and
weight_per_status(fed_tax_rel,WT_1) and
         X = N_1*WT_1.


the judgmt_sat_weight_val of judgmt_sat_info is X
 if the judgmt_sat_n of judgmt_sat_info is N_1 and
weight_per_status(judgmt_sat,WT_1) and
         X = N_1*WT_1.


the judg_vacat_weight_val of judg_vacat_info is X
 if
the judg_vacat_n of judg_vacat_info is N_1 and
weight_per_status(judg_vacat,WT_1) and
         X = N_1*WT_1.
```

```
the mech_lien_weight_val of mech_lien_info is X
      if
the mech_lien_n of mech_lien_info is N_1 and
weight_per_status(mech_lien,WT_1) and
         X = N_1*WT_1.


the mech_rele_weight_val of mech_rele_info is X
         if
the mech_rele_n of mech_rele_info is N_1 and
weight_per_status(mech_rele,WT_1) and
         X = N_1*WT_1.


the mn_mtg_fil_weight_val of mn_mtg_fil_info is X
 if
the mn_mtg_fil_n of mn_mtg_fil_info is N_1 and
weight_per_status(mn_mtg_fil,WT_1) and
         X = N_1*WT_1.


the nt_respon_weight_val of nt_respon_info is X
      if
the nt_respon_n of nt_respon_info is N_1 and
weight_per_status(nt_respon,WT_1) and
         X = N_1*WT_1.


the stat_tx_ln_weight_val of stat_tx_ln_info is X
 if
the stat_tx_ln_n of stat_tx_ln_info is N_1 and
weight_per_status(stat_tx_ln,WT_1) and
         X = N_1*WT_1.


the sta_tx_rel_weight_val of sta_tx_rel_info is X
 if
the sta_tx_rel_n of sta_tx_rel_info is N_1 and
weight_per_status(sta_tx_rel,WT_1) and
         X = N_1*WT_1.

the suit_dismd_weight_val of suit_dismd_info is X
 if
the suit_dismd_n of suit_dismd_info is N_1 and
weight_per_status(suit_dismd,WT_1) and
         X = N_1*WT_1.

 the wage_asign_weight_val of wage_asign_info is X
 if
```

```
the wage_asign_n of wage_asign_info is N_1 and
weight_per_status(wage_asign,WT_1) and
         X = N_1*WT_1.


the wa_release_weight_val of wa_release_info is X
 if
the wa_release_n of wa_release_info is N_1 and
weight_per_status(wa_release,WT_1) and
         X = N_1*WT_1.


the refinanced_weight_val of refinanced_info is X
 if
the refinanced_n of refinanced_info is N_1 and
weight_per_status(refinanced,WT_1) and
         X = N_1*WT_1.


the cr_cd_lost_weight_val of cr_cd_lost_info is X
 if
the cr_cd_lost_n of cr_cd_lost_info is N_1 and
weight_per_status(cr_cd_lost,WT_1) and
         X = N_1*WT_1.


the clos_inac_weight_val of clos_inac_info is X
     if
the clos_inac_n of clos_inac_info is N_1 and
weight_per_status(clos_inac,WT_1) and
         X = N_1*WT_1.


the transfered_weight_val of transfered_info is X
 if
the transfered_n of transfered_info is N_1 and
weight_per_status(transfered,WT_1) and
         X = N_1*WT_1.


the too_new_rt_weight_val of too_new_rt_info is X
 if
the too_new_rt_n of too_new_rt_info is N_1 and
weight_per_status(too_new_rt,WT_1) and
         X = N_1*WT_1.

the paid_satis_weight_val of paid_satis_info is X
 if
the paid_satis_n of paid_satis_info is N_1 and
```

```
weight_per_status(paid_satis,WT_1) and
         X = N_1*WT_1.


the paid_acct_weight_val of paid_acct_info is X
      if
the paid_acct_n of paid_acct_info is N_1 and
weight_per_status(paid_acct,WT_1) and
         X = N_1*WT_1.



the deceased_weight_val of deceased_info is X
     if
the deceased_n of deceased_info is N_1 and
weight_per_status(deceased,WT_1) and
         X = N_1*WT_1.

the cr_ln_clos_weight_val of cr_ln_clos_info is X
 if
the cr_ln_clos_n of cr_ln_clos_info is N_1 and
weight_per_status(cr_ln_clos,WT_1) and
         X = N_1*WT_1.


the redmd_repo_weight_val of redmd_repo_info is X
    if
the redmd_repo_n of redmd_repo_info is N_1 and
weight_per_status(redmd_repo,WT_1) and
         X = N_1*WT_1.

the cur_was_col_weight_val of cur_was_col_info is X
    if
the cur_was_col_n of cur_was_col_info is N_1 and
weight_per_status(cur_was_col,WT_1) and
         X = N_1*WT_1.


the cr_ln_rnst_weight_val of cr_ln_rnst_info is X
    if
the cr_ln_rnst_n of cr_ln_rnst_info is N_1 and
weight_per_status(cr_ln_rnst,WT_1) and
         X = N_1*WT_1.


the cur_was_for_weight_val of cur_was_for_info is X
    if
the cur_was_for_n of cur_was_for_info is N_1 and
weight_per_status(cur_was_for,WT_1) and
```

```
         X = N_1*WT_1.


the pd_not_aa_weight_val of pd_not_aa_info is X
      if
the pd_not_aa_n of pd_not_aa_info is N_1 and
weight_per_status(pd_not_aa,WT_1) and
         X = N_1*WT_1.


the city_tx_ln_weight_val of city_tx_ln_info is X
 if
the city_tx_ln_n of city_tx_ln_info is N_1 and
weight_per_status(city_tx_ln,WT_1) and
         X = N_1*WT_1.

 the city_tx_rel_weight_val of city_tx_rel_info is X
 if
the city_tx_rel_n of city_tx_rel_info is N_1 and
weight_per_status(city_tx_rel,WT_1) and
         X = N_1*WT_1.

the consel_ser_weight_val of consel_ser_info is X
 if
the consel_ser_n of consel_ser_info is N_1 and
weight_per_status(consel_ser,WT_1) and
         X = N_1*WT_1.


the co_tax_ln_weight_val of co_tax_ln_info is X
      if
the co_tax_ln_n of co_tax_ln_info is N_1 and
weight_per_status(co_tax_ln,WT_1) and
         X = N_1*WT_1.


the co_tax_rel_weight_val of co_tax_rel_info is X
if
the co_tax_rel_n of co_tax_rel_info is N_1 and
weight_per_status(co_tax_rel,WT_1) and
         X = N_1*WT_1.

the pd_foreclo_weight_val of pd_foreclo_info is X
if
the pd_foreclo_n of pd_foreclo_info is N_1 and
weight_per_status(pd_foreclo,WT_1) and
         X = N_1*WT_1.
```

```
the pd_chg_off_weight_val of pd_chg_off_info is X
if
the pd_chg_off_n of pd_chg_off_info is N_1 and
weight_per_status(pd_chg_off,WT_1) and
         X = N_1*WT_1.


the pd_repo_weight_val of pd_repo_info is X
     if
the pd_repo_n of pd_repo_info is N_1 and
weight_per_status(pd_repo,WT_1) and
         X = N_1*WT_1.


the pd_coll_ac_weight_val of pd_coll_ac_info is X
if
the pd_coll_ac_n of pd_coll_ac_info is N_1 and
weight_per_status(pd_coll_ac,WT_1) and
         X = N_1*WT_1.


the pacc_wdel_weight_val of pacc_wdel_info is X
     if
the pacc_wdel60_n of pacc_wdel60_info is N_1 and
weight_per_status(pacc_wdel60,WT_1) and
    Pacc_wdel60_WT = N_1*WT_1 and
the pacc_wdel90_n of pacc_wdel90_info is N_2 and
weight_per_status(pacc_wdel90,WT_2) and
    Pacc_wdel90_WT = N_2*WT_2 and
the pacc_wdel120_n of pacc_wdel120_info is N_3 and
weight_per_status(pacc_wdel120,WT_3) and
    Pacc_wdel120_WT = N_3*WT_3 and
the pacc_wdel150_n of pacc_wdel150_info is N_4 and
weight_per_status(pacc_wdel150,WT_4) and
    Pacc_wdel150_WT = N_4*WT_4 and
the pacc_wdel180_n of pacc_wdel180_info is N_5 and
weight_per_status(pacc_wdel180,WT_5) and
    Pacc_wdel180_WT = N_5*WT_5 and
X = Pacc_wdel60_WT + Pacc_wdel90_WT + Pacc_wdel120_WT +
Pacc_wdel150_WT + Pacc_wdel180_WT.


the suit_weight_val of suit_info is X
     if
the suit_n of suit_info is N_1 and
weight_per_status(suit,WT_1) and
          X = N_1*WT_1.

the pacc_wpd_weight_val of pacc_wpd_info is X
```

```
     if
the pacc_wpd_30_n of pacc_wpd_30_info is N_1 and
weight_per_status(pacc_wpd_30,WT_1) and
Pacc_wpd_30_WT = N_1*WT_1 and
the pacc_wpd_30by23_n of pacc_wpd_30by23_info is N_2
and
weight_per_status(pacc_wpd_30by23,WT_2) and
    Pacc_wpd_30by23_WT = N_2*WT_2 and
the pacc_wpd_30by4_n of pacc_wpd_30by4_info is N_4 and
weight_per_status(pacc_wpd_30by4,WT_4) and
    Pacc_wpd_30by4_WT = N_4*WT_4 and
the pacc_wpd_30by5_n of pacc_wpd_30by5_info is N_5 and
weight_per_status(pacc_wpd_30by5,WT_5) and
    Pacc_wpd_30by5_WT = N_5*WT_5 and
the pacc_wpd_30by6_n of pacc_wpd_30by6_info is N_6 and
weight_per_status(pacc_wpd_30by6,WT_6) and
    Pacc_wpd_30by6_WT = N_6*WT_6 and
X = Pacc_wpd_30_WT + Pacc_wpd_30by23_WT +
Pacc_wpd_30by4_WT +
Pacc_wpd_30by5_WT + Pacc_wpd_30by6_WT.

the accpd_weight_val of accpd_info is X
     if
the accpd_30_n of accpd_30_info is N_1 and
weight_per_status(accpd_30,WT_1) and
    Accpd_30_WT = N_1*WT_1 and
the accpd_30by2_n of accpd_30by2_info is N_2 and
weight_per_status(accpd_30by2,WT_2) and
    Accpd_30by2_WT = N_2*WT_2 and
the accpd_30by3_n of accpd_30by3_info is N_3 and
weight_per_status(accpd_30by3,WT_3) and
    Accpd_30by3_WT = N_3*WT_3 and
the accpd_30by4_n of accpd_30by4_info is N_4 and
weight_per_status(accpd_30by4,WT_4) and
    Accpd_30by4_WT = N_4*WT_4 and
the accpd_30by5_n of accpd_30by5_info is N_5 and
weight_per_status(accpd_30by5,WT_5) and
    Accpd_30by5_WT = N_5*WT_5 and
the accpd_30by6_n of accpd_30by6_info is N_6 and
weight_per_status(accpd_30by6,WT_6) and
    Accpd_30by6_WT = N_6*WT_6 and
X = Accpd_30_WT + Accpd_30by2_WT + Accpd_30by3_WT +
Accpd_30by4_WT + Accpd_30by5_WT + Accpd_30by6_WT.

the cwpd_weight_val of curwaspd_info is X
     if
the cwpd_n of cwpd_info is N_1 and
weight_per_status(cwpd,WT_1) and
    Cwpd_WT = N_1*WT_1 and
```

```
the cwpd_30_n of cwpd_30_info is N_2 and
weight_per_status(cwpd_30,WT_2) and
    Cwpd_30_WT = N_2*WT_2 and
the cwpd_30by2_n of cwpd_30by2_info is N_3 and
weight_per_status(cwpd_30by2,WT_3) and
    Cwpd_30by2_WT = N_3*WT_3 and
the cwpd_30by3_n of cwpd_30by3_info is N_4 and
weight_per_status(cwpd_30by3,WT_4) and
    Cwpd_30by3_WT = N_4*WT_4 and
the cwpd_30by4_n of cwpd_30by4_info is N_5 and
weight_per_status(cwpd_30by4,WT_5) and
    Cwpd_30by4_WT = N_5*WT_5 and
the cwpd_30by5_n of cwpd_30by5_info is N_6 and
weight_per_status(cwpd_30by5,WT_6) and
    Cwpd_30by5_WT = N_6*WT_6 and
the cwpd_30by6_n of cwpd_30by6_info is N_7 and
weight_per_status(cwpd_30by6,WT_7) and
    Cwpd_30by6_WT = N_7*WT_7 and
X = Cwpd_WT + Cwpd_30_WT + Cwpd_30by2_WT +
Cwpd_30by3_WT +
Cwpd_30by4_WT + Cwpd_30by5_WT + Cwpd_30by6_WT.


the delinq_weight_val of delinq_info is X
     if
the d_60_n of d_60_info is N_1 and
weight_per_status(delinq_60,WT_1) and
    D_60_WT = N_1*WT_1 and
the d_90_n of d_90_info is N_2 and
weight_per_status(delinq_90,WT_2) and
    D_90_WT = N_2*WT_2 and
the d_120_n of d_120_info is N_3 and
weight_per_status(delinq_120,WT_3) and
    D_120_WT = N_3*WT_3 and
the d_150_n of d_150_info is N_4 and
weight_per_status(delinq_150,WT_4) and
    D_150_WT = N_4*WT_4 and
the d_180_n of d_180_info is N_5 and
weight_per_status(delinq_180,WT_5) and
    D_180_WT = N_5*WT_5 and
X = D_60_WT + D_90_WT + D_120_WT + D_150_WT + D_180_WT.


the cur_was_weight_val of cur_was_info is X
     if
the c_w_60_n of c_w_60_info is N_1 and
weight_per_status(c_w_60,WT_1) and
    C_w_60_WT = N_1*WT_1 and
the c_w_90_n of c_w_90_info is N_2 and
```

```
weight_per_status(c_w_90,WT_2) and
    C_w_90_WT = N_2*WT_2 and
the c_w_120_n of c_w_120_info is N_3 and
weight_per_status(c_w_120,WT_3) and
    C_w_120_WT = N_3*WT_3 and
the c_w_150_n of c_w_150_info is N_4 and
weight_per_status(c_w_150,WT_4) and
    C_w_150_WT = N_4*WT_4 and
the c_w_180_n of c_w_180_info is N_5 and
weight_per_status(c_w_180,WT_5) and
    C_w_180_WT = N_5*WT_5 and
X = C_w_60_WT + C_w_90_WT + C_w_120_WT +
    C_w_150_WT + C_w_180_WT.


the pdbydlr_weight_val of pdbydlr_info is X
     if the pdbydlr_n of pdbydlr_info is N_1 and

weight_per_status(pdbydlr,WT_1) and
          X = N_1*WT_1.


the coll_acct_weight_val of coll_acct_info is X
     if
the coll_acct_n of coll_acct_info is N and
weight_per_status(coll_acct,WT) and
X = N*WT.


the charge_off_weight_val of charge_off_info is X
if
the charge_off_n of charge_off_info is N and
weight_per_status(charge_off,WT) and
X = N*WT.

the inq_weight_val of inquiry_info is X
     if
the no_of_inq of inquiry_info is N and
weight_per_status(inquiry,WT) and
X = N*WT.

the curr_acct_n_weight_val of superpayer_info is X
if
the curr_acct_n of superpayer_info is N and
weight_per_status(curr_acct,WT) and
X = N*WT.
the superpayer_weight_val of superpayer_info is X
if
the monthly_payment of the superpayer_info is E and
the monthly_income of the personal_info is I and
E > I and weight_for_superpayer(X).

the superpayer_weight_val of superpayer_info is X
if
the monthly_payment of the superpayer_info is E and
the monthly_income of the personal_info is I and
E < I and
X = 0.

monthly_payment of superpayer_info is E
     if
ssn of personal_info is SSN and
compute_total_monthly_payment(SSN,E).



monthly_income of personal_info is I
   if
ssn of personal_info is SSN and
compute_discretionary_income(SSN,I).


the insclaim_weight_val of insclaim_info is X
     if
the insclaim_n of insclaim_info is N_1 and
weight_per_status(insclaim,WT_1) and
X = N_1*WT_1.

the notpdaa_weight_val of notpdaa_info is X
     if
the notpdaa_n of notpdaa_info is N_1 and
weight_per_status(notpdaa,WT_1) and
          X = N_1*WT_1.

the volnrepo_weight_val of volnrepo_info is X
     if
the volnrepo_n of volnrepo_info is N_1 and
weight_per_status(volnrepo,WT_1) and
          X = N_1*WT_1..



    /* CRD13.TAX  (TAXONOMY)  How the knowledge is
represented*/
type weight_pt_info = role.
type cut_off_info = role.
type customer_profile_info = role.
type superpayer_info = role.
type bankrupt_info = role.
type currency_info = role.
type bk_already_info = role.

type trw_info = role.
type individual_info = role.

type conclusion_info = role.

type personal_info = role.
type inquiry_info = role.
type weight_pt_val =
[extremely_low,very_low,low,moderately_high,
     high,very_high,extremely_high,alarmingly_high].
type pb_cut_off_val = numeric.
type extremely_high_cut_off_val = numeric.
type very_high_cut_off_val = numeric.
type high_cut_off_val = numeric.
type moderately_high_cut_off_val = numeric.
type low_cut_off_val = numeric.
type very_low_cut_off_val = numeric.
type extremely_low_cut_off_val = numeric.

type monthly_payment = numeric.
type monthly_income = numeric.
type ssn = numeric.
type sal_step = numeric.
type zip = numeric.
type m_status = [single,married,divorced].
type dob = numeric.
type concluded =
[too_many_accts,superpayer,normal,satisfactory,
poor,very_poor,serious,
critical,potential_bankrupt,very_critical].
type curr_acct_n = numeric.
type curr_acct_n_weight_val = numeric.
type cur_was_vals = [60,90,120,150,180,for,col]. type

paid_vals =
[90,120,150,180,collac,repo,chgoff,foreclo,bydler].

type delinq_vals = [60,90,120,150,180].
type delinq_was_vals = [90,120].
type bk_vals =
     [bk_7_file,bk_7_disc,bk_7_dism,bk_11_file,
     bk_1i_disc,bk_11_dism,bk_13_file,bk_13_dism,
     bk_13_comp,not_bankrupt].
```

```
type conc_bank = [bk_red,bk_green,bk_yellow,bk_orange].

type curr_acct_info = role.
type cur_was_info = role.
type delinq_info = role.

type paid_info = role.
type coll_acct_info = role.
type x_30_day_del_info = role.
type pd_coll_ac_info = role.
type charge_off_info = role.
type bk_liq_reo_info = role.
type foreclosure_info = role.
type judgment_info = role.
type repo_info = role.
type pdbydlr_info = role.
type no_of_inq = numeric.
type c_w_60_info = role.
type c_w_90_info = role.
type c_w_120_info = role.
type c_w_150_info = role.
type c_w_180_info = role.
type c_w_60_n = numeric.
type c_w_60_a = numeric.
type c_w_90_n = numeric.
type c_w_90_a = numeric.
type c_w_120_n = numeric.
type c_w_120_a = numeric.
type c_w_150_n = numeric.
type c_w_150_a = numeric.
type c_w_180_n = numeric.
type c_w_180_a = numeric.
type d_60_info = role.
type d_90_info = role.
type d_120_info = role.
type d_150_info = role.
type d_180_info = role.
type d_60_n = numeric.
type d_60_a = numeric.
type d_90_n = numeric.
type d_90_a = numeric.
type d_120_n = numeric.
type d_120_a = numeric.
type d_150_n = numeric.
type d_150_a = numeric.
type d_180_n = numeric.
type d_180_a = numeric.
type pdbydlr_n = numeric.
type pdbydlr_a = numeric.
type debt_equity_ratio = numeric.
type bk_liq_rep_n = numeric.
type coll_acct_n = numeric.
type coll_acct_a = numeric.
type x_30_day_del_n = numeric.
type x_30_day_del_a = numeric.
type curr_acct_n = numeric.
type overall_weight_val = numeric.
type public_record_val = numeric.
type private_record_val = numeric.
type set_of_bkrpts_and_liens_weight_val = numeric.
type set_of_misc_public_items_weight_val = numeric.
type set_of_current_and_paid_accts_weight_val =
    numeric.
type set_of_delinq_accts_weight_val = numeric.
type set_of_misc_private_items_weight_val = numeric.
type delinq_weight_val = numeric.
type cur_was_weight_val = numeric.
type coll_acct_weight_val = numeric.
type pd_coll_ac_n = numeric.
type pd_coll_ac_a = numeric.
type pd_coll_ac_weight_val = numeric.
type charge_off_n = numeric.
type charge_off_a = numeric.
type charge_off_weight_val = numeric.
type inq_weight_val = numeric.
type superpayer_weight_val = numeric.
type foreclosure_weight_val = numeric.
type judgment_weight_val = numeric.
type repo_weight_val = numeric.
type pdbydlr_weight_val = numeric.
type insclaim_info = role.
type insclaim_n = numeric.
type insclaim_a = numeric.
type insclaim_weight_val = numeric.
type notpdaa_info = role.
type notpdaa_n = numeric.
type notpdaa_a = numeric.
type notpdaa_weight_val = numeric.
type volnrepo_info = role.
type volnrepo_n = numeric.
type volnrepo_a = numeric.
type volnrepo_weight_val = numeric.
type cwpd_weight_val = numeric.
type curwaspd_info = role.
type cwpd_info = role.
type cwpd_30_info = role.
type cwpd_30by2_info = role.
type cwpd_30by3_info = role.
type cwpd_30by4_info = role.
type cwpd_30by5_info = role.
type cwpd_30by6_info = role.
type cwpd_n = numeric.
type cwpd_a = numeric.
type cwpd_30_n = numeric.
type cwpd_30_a = numeric.
type cwpd_30by2_n = numeric.
type cwpd_30by2_a = numeric.
type cwpd_30by3_n = numeric.
type cwpd_30by3_a = numeric.
type cwpd_30by4_n = numeric.
type cwpd_30by4_a = numeric.
type cwpd_30by5_n = numeric.
type cwpd_30by5_a = numeric.
type cwpd_30by6_n = numeric.
type cwpd_30by6_a = numeric.
type accpd_weight_val = numeric.
type accpd_info = role.
type accpd_30_info = role.
type accpd_30by2_info = role.
type accpd_30by3_info = role.
type accpd_30by4_info = role.
type accpd_30by5_info = role.
type accpd_30by6_info = role.
type accpd_30_n = numeric.
type accpd_30_a = numeric.
type accpd_30by2_n = numeric.
type accpd_30by2_a = numeric.
type accpd_30by3_n = numeric.
type accpd_30by3_a = numeric.
type accpd_30by4_n = numeric.
type accpd_30by4_a = numeric.
type accpd_30by5_n = numeric.
type accpd_30by5_a = numeric.
type accpd_30by6_n = numeric.
type accpd_30by6_a = numeric.
type pacc_wpd_info = role.
type pacc_wpd_30_info = role.
type pacc_wpd_30by23_info = role.
type pacc_wpd_30by4_info = role.
type pacc_wpd_30by5_info = role.
type pacc_wpd_30by6_info = role.
type pacc_wpd_30_n = numeric.
type pacc_wpd_30_a = numeric.
type pacc_wpd_30by23_n = numeric.
type pacc_wpd_30by23_a = numeric.
type pacc_wpd_30by4_n = numeric.
type pacc_wpd_30by4_a = numeric.
type pacc_wpd_30by5_n = numeric.
type pacc_wpd_30by5_a = numeric.
type pacc_wpd_30by6_n = numeric.
type pacc_wpd_30by6_a = numeric.
type pacc_wpd_weight_val = numeric.
type suit_info = role.
type suit_n = numeric.
type suit_a = numeric.
type suit_weight_val = numeric.
type pacc_wdel_info = role.
type pacc_wdel60_info = role.
type pacc_wdel90_info = role.
type pacc_wdel120_info = role.
type pacc_wdel150_info = role.
type pacc_wdel180_info = role.
type pacc_wdel60_n = numeric.
type pacc_wdel60_a = numeric.
type pacc_wdel90_n = numeric.
type pacc_wdel90_a = numeric.
type pacc_wdel120_n = numeric.
type pacc_wdel120_a = numeric.
type pacc_wdel150_n = numeric.
type pacc_wdel150_a = numeric.
type pacc_wdel180_n = numeric.
type pacc_wdel180_a = numeric.
type pacc_wdel_weight_val = numeric.
type pd_repo_info = role.
type pd_repo_n = numeric.
type pd_repo_a = numeric.
type pd_repo_weight_val = numeric.
type pd_chg_off_info = role.
type pd_chg_off_n = numeric.
type pd_chg_off_a = numeric.
type pd_chg_off_weight_val = numeric.
type pd_foreclo_info = role.
type pd_foreclo_n = numeric.
type pd_foreclo_a = numeric.
type pd_foreclo_weight_val = numeric.
type bkliqreo_info = role.
type bkliqreo_n = numeric.
type bkliqreo_a = numeric.
type bkliqreo_weight_val = numeric.
type settled_info = role.
type settled_n = numeric.
type settled_a = numeric.
type settled_weight_val = numeric.
type bk_adj_pln_info = role.
type bk_adj_pln_n = numeric.
type bk_adj_pln_a = numeric.
type bk_adj_pln_weight_val = numeric.
type scnl_nwloc_info = role.
type scnl_nwloc_n = numeric.
type scnl_nwloc_a = numeric.
type scnl_nwloc_weight_val = numeric.
type co_now_pay_info = role.
type co_now_pay_n = numeric.
type co_now_pay_a = numeric.
type co_now_pay_weight_val = numeric.
type fore_proc_info = role.
type fore_proc_n = numeric.
type fore_proc_a = numeric.
type fore_proc_weight_val = numeric.
type gov_claim_info = role.
type gov_claim_n = numeric.
type gov_claim_a = numeric.
type gov_claim_weight_val = numeric.
type close_np_aa_info = role.
type close_np_aa_n = numeric.
type close_np_aa_a = numeric.
type close_np_aa_weight_val = numeric.
type scnl_info = role.
type scnl_n = numeric.
type scnl_a = numeric.
type scnl_weight_val = numeric.
type fed_tax_ln_info = role.
type fed_tax_ln_n = numeric.
type fed_tax_ln_a = numeric.
type fed_tax_ln_weight_val = numeric.
type fed_tax_rel_info = role.
type fed_tax_rel_n = numeric.
type fed_tax_rel_a = numeric.
type fed_tax_rel_weight_val = numeric.
type judgmt_sat_info = role.
type judgmt_sat_n = numeric.
type judgmt_sat_a = numeric.
type judgmt_sat_weight_val = numeric.
type judg_vacat_info = role.
type judg_vacat_n = numeric.
type judg_vacat_a = numeric.
type judg_vacat_weight_val = numeric.
type mech_lien_info = role.
type mech_lien_n = numeric.
type mech_lien_a = numeric.
type mech_lien_weight_val = numeric.
type mech_rele_info = role.
type mech_rele_n = numeric.
type mech_rele_a = numeric.
type mech_rele_weight_val = numeric.
type mn_mtg_fil_info = role.
type mn_mtg_fil_n = numeric.
type mn_mtg_fil_a = numeric.
type mn_mtg_fil_weight_val = numeric.
type nt_respon_info = role.
type nt_respon_n = numeric.
type nt_respon_a = numeric.
type nt_respon_weight_val = numeric.
type stat_tx_ln_info = role.
type stat_tx_ln_n = numeric.
type stat_tx_ln_a = numeric.
type stat_tx_ln_weight_val = numeric.
type sta_tx_rel_info = role.
type sta_tx_rel_n = numeric.
type sta_tx_rel_a = numeric.
type sta_tx_rel_weight_val = numeric.
type suit_dismd_info = role.
type suit_dismd_n = numeric.
type suit_dismd_a = numeric.
type suit_dismd_weight_val = numeric.
type wage_asign_info = role.
type wage_asign_n = numeric.
type wage_asign_a = numeric.
type wage_asign_weight_val = numeric.
type wa_release_info = role.
type wa_release_n = numeric.
type wa_release_a = numeric.
type wa_release_weight_val = numeric.
type refinanced_info = role.
type refinanced_n = numeric.
type refinanced_a = numeric.
type refinanced_weight_val = numeric.
type cr_cd_lost_info = role.
type cr_cd_lost_n = numeric.
type cr_cd_lost_a = numeric.
type cr_cd_lost_weight_val = numeric.
type clos_inac_info = role.
clos_inac_n = numeric.
type clos_inac_a = numeric.
type clos_inac_weight_val = numeric.
type transfered_info = role.
type transfered_n = numeric.
type transfered_a = numeric.
type transfered_weight_val = numeric.
type too_new_rt_info = role.
type too_new_rt_n = numeric.
type too_new_rt_a = numeric.
type too_new_rt_weight_val = numeric.
type paid_satis_info = role.
type paid_satis_n = numeric.
type paid_satis_a = numeric.
type paid_satis_weight_val = numeric.
type paid_acct_info = role.
type paid_acct_n = numeric.
type paid_acct_a = numeric.
type paid_acct_weight_val = numeric.
type cr_ln_clos_info = role.
type cr_ln_clos_n = numeric.
type cr_ln_clos_a = numeric.
type cr_ln_clos_weight_val = numeric.
type deceased_info = role.
type deceased_n = numeric.
type deceased_a = numeric.
type deceased_weight_val = numeric.
type redmd_repo_info = role.
type redmd_repo_n = numeric.
type redmd_repo_a = numeric.
type redmd_repo_weight_val = numeric.
type cur_was_col_info = role.
type cur_was_col_n = numeric.
type cur_was_col_a = numeric.
type cur_was_col_weight_val = numeric.
type cr_ln_rnst_info = role.
type cr_ln_rnst_n = numeric.
type cr_ln_rnst_a = numeric.
type cr_ln_rnst_weight_val = numeric.
type cur_was_for_info = role.
type cur_was_for_n = numeric.
type cur_was_for_a = numeric.

type cur_was_for_weight_val = numeric.

type pd_not_aa_info = role.
type pd_not_aa_n = numeric.
type pd_not_aa_a = numeric.
type pd_not_aa_weight_val = numeric.
type city_tx_ln_info = role.
type city_tx_ln_n = numeric.
type city_tx_ln_a = numeric.
type city_tx_ln_weight_val = numeric.
type city_tx_rel_info = role.
type city_tx_rel_n = numeric.
type city_tx_rel_a = numeric.
type city_tx_rel_weight_val = numeric.
type consel_ser_info = role.
type consel_ser_n = numeric.
type consel_ser_a = numeric.
type consel_ser_weight_val = numeric.
type co_tax_ln_info = role.
type co_tax_ln_n = numeric.
type co_tax_ln_a = numeric.
type co_tax_ln_weight_val = numeric.
type co_tax_rel_info = role.
type co_tax_rel_n = numeric.
type co_tax_rel_a = numeric.
type co_tax_rel_weight_val = numeric.


define primitive perserec with
   customer_profile_info = customer_profile and
   superpayer_info = superpayer and
   bankrupt_info = bankrupt and
   weight_pt_info = weight_pt and
   cut_off_info = cut_off and
   conclusion_info = conclusion.

define primitive customer_profile with
   personal_info = personal and
   trw_info = trw.

define primitive superpayer with
   debt_equity_ratio = (0,3) and
   curr_acct_n = (0,100) and
 curr_acct_n_weight_val = (0,1000) and
 monthly_payment = (0,20000) and
   superpayer_weight_val = (0,1000).

define primitive bankrupt with
   conc_bank = [bk_red,bk_green,bk_yellow,bk_orange].

define primitive weight_pt with
     weight_pt_val =
[extremely_low,very_low,low,moderately_high,
high,very_high,extremely_high,alarmingly_high].

define primitive cut_off with
     pb_cut_off_val = (399,401) and
     extremely_high_cut_off_val = (200,250) and
     very_high_cut_off_val = (170,199) and
     high_cut_off_val = (120,149) and
     moderately_high_cut_off_val = (100,119) and
     low_cut_off_val = (80,99) and
     very_low_cut_off_val = (40,59) and
     extremely_low_cut_off_val = (0,39).

define primitive conclusion with
   concluded =
      [too_many_accts,superpayer,normal,satisfactory,
      poor,very_poor,serious,
      critical,potential_bankrupt,very_critical]
           and
          overall_weight_val = (0,10000) and
          public_record_val = (0,10000) and
          private_record_val = (0,10000) and
          set_of_bkrpts_and_liens_weight_val =
        (0,10000) and
          set_of_misc_public_items_weight_val =
        (0,10000) and
          set_of_current_and_paid_accts_weight_val =
        (0,10000) and
          set_of_delinq_accts_weight_val = (0,10000)
        and
          set_of_misc_private_items_weight_val =
        (0,10000).

define primitive personal with
  ssn = (1,999999999) and
   zip = (10000,99999) and
   m_status = [single,married,divorced] and
   dob = (1900,1986) and
   monthly_income = (0,20000).


define primitive trw with
   curr_acct_info = curr_acct and
   cur_was_info = cur_was and
   delinq_info = delinq and
   pdbydlr_info = pdbydlr and
   coll_acct_info = coll_acct and
   x_30_day_del_info = x_30_day_del and
   bk_liq_reo_info = bk_liq_reo and
   bk_vals =
[bk_7_file,bk_7_disc,bk_7_dism,bk_11_file,bk_11_disc,
bk_11_dism,bk_13_file,bk_13_dism,bk_13_com-
p,not_bankrupt]
   and
   pd_coll_ac_info = pd_coll_ac and
   charge_off_info = charge_off and
   foreclosure_info = foreclosure  and
   judgment_info = judgment and
   repo_info = repo and
   inquiry_info = inquiry and
```

```
insclaim_info = insclaim and
notpdaa_info = notpdaa and
volnrepo_info = volnrepo and
curwaspd_info = curwaspd and
accpd_info = accpd and
pacc_wpd_info = pacc_wpd and
suit_info = suit and
pd_repo_info = pd_repo and
pd_chg_off_info = pd_chg_off and
pd_foreclo_info = pd_foreclo and
bkliqreo_info = bkliqreo and
pacc_wdel_info = pacc_wdel and
settled_info = settled and
bk_adj_pln_info = bk_adj_pln and
scnl_nwloc_info = scnl_nwloc and
co_now_pay_info = co_now_pay and
fore_proc_info = fore_proc and
gov_claim_info = gov_claim and
close_np_aa_info = close_np_aa and
scnl_info = scnl and
fed_tax_ln_info = fed_tax_ln and
fed_tax_rel_info = fed_tax_rel and
judgmt_sat_info = judgmt_sat and
judg_vacat_info = judg_vacat and
mech_lien_info = mech_lien and
mech_rele_info = mech_rele and
 mn_mtg_fil_info = mn_mtg_fil and
 nt_respon_info = nt_respon and
stat_tx_ln_info = stat_tx_ln and
sta_tx_rel_info = sta_tx_rel and
suit_dismd_info = suit_dismd and
wage_asign_info = wage_asign and
wa_release_info = wa_release and
refinanced_info = refinanced and
cr_cd_lost_info = cr_cd_lost and
clos_inac_info = clos_inac and
transfered_info = transfered and
too_new_rt_info = too_new_rt and
paid_satis_info = paid_satis and
paid_acct_info = paid_acct and
cr_ln_clos_info = cr_ln_clos and
deceased_info = deceased and
redmd_repo_info = redmd_repo and
cur_was_col_info = cur_was_col and
cr_ln_rnst_info = cr_ln_rnst and
cur_was_for_info = cur_was_for and
pd_not_aa_info = pd_not_aa and
city_tx_ln_info = city_tx_ln and
city_tx_rel_info = city_tx_rel and
```

```
   consel_ser_info = consel_ser and
   co_tax_ln_info = co_tax_ln and
   co_tax_rel_info = co_tax_rel.


define primitive pacc_wdel with
   pacc_wdel60_info = pacc_wdel60 and
   pacc_wdel90_info = pacc_wdel90 and
   pacc_wdel120_info = pacc_wdel120 and
   pacc_wdel150_info = pacc_wdel150 and
   pacc_wdel180_info = pacc_wdel180 and
   pacc_wdel_weight_val = (0,1000).

define primitive pacc_wdel60 with
   pacc_wdel60_n = (0,100) and
   pacc_wdel60_a = (0,10000).


define primitive pacc_wdel90 with
    pacc_wdel90_n = (0,100) and
    pacc_wdel90_a = (0,10000).


define primitive pacc_wdel120 with
   pacc_wdel120_n = (0,100) and
   pacc_wdel120_a = (0,10000).


define primitive pacc_wdel150 with
   pacc_wdel150_n = (0,100) and
   pacc_wdel150_a = (0,10000).


define primitive pacc_wdel180 with
   pacc_wdel180_n = (0,100) and
   pacc_wdel180_a = (0,10000).


define primitive suit with
   suit_n = (0,100) and
   suit_a = (0,10000) and
   suit_weight_val = (0,1000).


define primitive cur_was with
   c_w_60_info = c_w_60 and
   c_w_90_info = c_w_90 and
   c_w_120_info = c_w_120 and
   c_w_150_info = c_w_150 and
   c_w_180_info = c_w_180 and
   cur_was_weight_val = (0,1000).

define primitive c_w_60 with
  c_w_60_n = (0,100) and
  c_w_60_a = (0,10000).

define primitive c_w_90 with
   c_w_90_n = (0,100) and
   c_w_90_a = (0,10000).

define primitive c_w_120 with
   c_w_120_n = (0,100) and
   c_w_120_a = (0,10000).


define primitive c_w_150 with
  c_w_150_n = (0,100) and
  c_w_150_a = (0,10000).

define primitive c_w_180 with
   c_w_180_n = (0,100) and

   c_w_180_a = (0,10000).


define primitive delinq with
   d_60_info = d_60 and
   d_90_info = d_90 and
   d_120_info = d_120 and
   d_150_info = d_150 and
   d_180_info = d_180 and
   delinq_weight_val = (0,1000).



define primitive d_60 with
   d_60_n = (0,100) and
   d_60_a = (0,10000).


define primitive d_90 with
   d_90_n = (0,100) and
   d_90_a = (0,10000).

define primitive d_120 with
   d_120_n = (0,100) and
   d_120_a = (0,10000).


define primitive d_150 with
   d_150_n = (0,100) and
   d_150_a = (0,10000).


define primitive d_180 with
   d_180_n = (0,100) and
   d_180_a = (0,10000).


define primitive coll_acct with
   coll_acct_n = (0,100) and
   coll_acct_a = (0,10000) and
  coll_acct_weight_val = (0,1000).

define primitive x_30_day_del with
   x_30_day_del_n = (0,100) and
   x_30_day_del_a = (0,10000).

define primitive bk_liq_reo with
     bk_liq_reo_n = (0,100).


define primitive curr_acct with
     curr_acct_n = (0,100).

define primitive pd_coll_ac with
     pd_coll_ac_n = (0,100) and
     pd_coll_ac_a = (0,10000) and
     pd_coll_ac_weight_val = (0,1000).

define primitive charge_off with
     charge_off_n = (0,100) and
     charge_off_a = (0,10000) and
     charge_off_weight_val = (0,1000).

define primitive inquiry with
     no_of_inq = (0,100) and
     inq_weight_val = (0,1000).

define primitive foreclosure with
     foreclosure_weight_val = (0,1000).

 define primitive judgment with
     judgment_weight_val = (0,1000).
```

```
define primitive repo with
     repo_weight_val = (0,1000).

define primitive pdbydlr with
   pdbydlr_n = (0,100) and
   pdbydlr_a = (0,100000) and
   pdbydlr_weight_val = (0,1000).

define primitive insclaim with
   insclaim_n = (0,100) and
   insclaim_a = (0,100000) and
   insclaim_weight_val = (0,1000).

define primitive notpdaa with
   notpdaa_n = (0,100) and
   notpdaa_a = (0,100000) and
   notpdaa_weight_val = (0,1000).

define primitive volnrepo with
  volnrepo_n = (0,100) and
 volnrepo_a = (0,100000) and
  volnrepo_weight_val = (0,1000).


define primitive curwaspd with
   cwpd_info = cwpd and
   cwpd_30_info = cwpd_30 and
   cwpd_30by2_info = cwpd_30by2 and
   cwpd_30by3_info = cwpd_30by3 and
   cwpd_30by4_info = cwpd_30by4 and
   cwpd_30by5_info = cwpd_30by5 and
   cwpd_30by6_info = cwpd_30by6 and
   cwpd_weight_val = (0,1000).

define primitive cwpd with
   cwpd_n = (0,100) and
   cwpd_a = (0,10000).

define primitive cwpd_30 with
  cwpd_30_n = (0,100) and
  cwpd_30_a = (0,10000).

define primitive cwpd_30by2 with
   cwpd_30by2_n = (0,100) and
   cwpd_30by2_a = (0,10000).

define primitive cwpd_30by3 with
   cwpd_30by3_n = (0,100) and
   cwpd_30by3_a = (0,10000).


define primitive cwpd_30by4 with
   cwpd_30by4_n = (0,100) and
   cwpd_30by4_a = (0,10000).


define primitive cwpd_30by5 with
   cwpd_30by5_n = (0,100) and
   cwpd_30by5_a = (0,10000).


define primitive cwpd_30by6 with
   cwpd_30by6_n = (0,100) and
   cwpd_30by6_a = (0,10000).


define primitive accpd with
   accpd_30_info = accpd_30 and
   accpd_30by2_info = accpd_30by2 and
   accpd_30by3_info = accpd_30by3 and
   accpd_30by4_info = accpd_30by4 and
   accpd_30by5_info = accpd_30by5 and
   accpd_30by6_info = accpd_30by6 and
   accpd_weight_val = (0,1000).

define primitive accpd_30 with
  accpd_30_n = (0,100) and
  accpd_30_a = (0,10000).


define primitive accpd_30by2 with

  accpd_30by2_n = (0,100) and

accpd_30by2_a = (0,10000).


define primitive accpd_30by3 with

  accpd_30by3_n = (0,100) and

accpd_30by3_a = (0,10000).


define primitive accpd_30by4 with
  accpd_30by4_n = (0,100) and
```

```
accpd_30by4_a = (0,10000).


define primitive accpd_30by5 with
  accpd_30by5_n = (0,100) and
  accpd_30by5_a = (0,10000).


define primitive accpd_30by6 with
  accpd_30by6_n = (0,100) and
  accpd_30by6_a = (0,10000).


define primitive pacc_wpd with
   pacc_wpd_30_info = pacc_wpd_30 and
   pacc_wpd_30by23_info = pacc_wpd_30by23 and
   pacc_wpd_30by4_info = pacc_wpd_30by4 and
   pacc_wpd_30by5_info = pacc_wpd_30by5 and
   pacc_wpd_30by6_info = pacc_wpd_30by6 and
   pacc_wpd_weight_val = (0,1000).

define primitive pacc_wpd_30 with
  pacc_wpd_30_n = (0,100) and
  pacc_wpd_30_a = (0,10000).


define primitive pacc_wpd_30by23 with
   pacc_wpd_30by23_n = (0,100) and
   pacc_wpd_30by23_a = (0,10000).


define primitive pacc_wpd_30by4 with
   pacc_wpd_30by4_n = (0,100) and
   pacc_wpd_30by4_a = (0,10000).


define primitive pacc_wpd_30by5 with
   pacc_wpd_30by5_n = (0,100) and
   pacc_wpd_30by5_a = (0,10000).


define primitive pacc_wpd_30by6 with
   pacc_wpd_30by6_n = (0,100) and
   pacc_wpd_30by6_a = (0,10000).


define primitive pd_repo with
   pd_repo_n = (0,100) and
   pd_repo_a = (0,10000) and
   pd_repo_weight_val = (0,1000).

define primitive pd_chg_off with
   pd_chg_off_n = (0,100) and

pd_chg_off_a = (0,10000) and
   pd_chg_off_weight_val = (0,1000).

define primitive pd_foreclo with
   pd_foreclo_n = (0,100) and

pd_foreclo_a = (0,10000) and
   pd_foreclo_weight_val = (0,1000).

define primitive bkliqreo with
   bkliqreo_n = (0,100) and

bkliqreo_a = (0,10000) and
   bkliqreo_weight_val = (0,1000).

define primitive settled with
   settled_n = (0,100) and
   settled_a = (0,10000) and
  settled_weight_val = (0,1000).

define primitive bk_adj_pln with
   bk_adj_pln_n = (0,100) and
   bk_adj_pln_a = (0,10000) and
   bk_adj_pln_weight_val = (0,1000).

define primitive scnl_nwloc with
   scnl_nwloc_n = (0,100) and
   scnl_nwloc_a = (0,10000) and
   scnl_nwloc_weight_val = (0,1000).

define primitive co_now_pay with
   co_now_pay_n = (0,100) and
   co_now_pay_a = (0,10000) and
   co_now_pay_weight_val = (0,1000).

define primitive fore_proc with
   fore_proc_n = (0,100) and
   fore_proc_a = (0,10000) and
   fore_proc_weight_val = (0,1000).
```

```
define primitive gov_claim with
   gov_claim_n = (0,100) and
   gov_claim_a = (0,10000) and
   gov_claim_weight_val = (0,1000).

define primitive close_np_aa with
   close_np_aa_n = (0,100) and
   close_np_aa_a = (0,10000) and
   close_np_aa_weight_val = (0,1000).


define primitive scnl with
   scnl_n = (0,100) and
   scnl_a = (0,10000) and
   scnl_weight_val = (0,1000).

define primitive fed_tax_ln with
   fed_tax_ln_n = (0,100) and
   fed_tax_ln_a = (0,10000) and
   fed_tax_ln_weight_val = (0,1000).

define primitive fed_tax_rel with
   fed_tax_rel_n = (0,100) and
   fed_tax_rel_a = (0,10000) and
   fed_tax_rel_weight_val = (0,1000).

define primitive judgmt_sat with
   judgmt_sat_n = (0,100) and
   judgmt_sat_a = (0,10000) and
   judgmt_sat_weight_val = (0,1000).

define primitive judg_vacat with
   judg_vacat_n = (0,100) and
   judg_vacat_a = (0,10000) and
   judg_vacat_weight_val = (0,1000).

define primitive mech_lien with
   mech_lien_n = (0,100) and
   mech_lien_a = (0,10000) and
   mech_lien_weight_val = (0,1000).

define primitive mech_rele with
   mech_rele_n = (0,100) and
   mech_rele_a = (0,10000) and
   mech_rele_weight_val = (0,1000).

define primitive mn_mtg_fil with
   mn_mtg_fil_n = (0,100) and
   mn_mtg_fil_a = (0,10000) and
   mn_mtg_fil_weight_val = (0,1000).

define primitive nt_respon with
   nt_respon_n = (0,100) and
   nt_respon_a = (0,10000) and
   nt_respon_weight_val = (0,1000).

define primitive stat_tx_ln with
   stat_tx_ln_n = (0,100) and
   stat_tx_ln_a = (0,10000) and
   stat_tx_ln_weight_val = (0,1000).

define primitive sta_tx_rel with
   sta_tx_rel_n = (0,100) and
   sta_tx_rel_a = (0,10000) and
   sta_tx_rel_weight_val = (0,1000).

define primitive suit_dismd with
   suit_dismd_n = (0,100) and
   suit_dismd_a = (0,10000) and
   suit_dismd_weight_val = (0,1000).

define primitive wage_asign with
   wage_asign_n = (0,100) and
   wage_asign_a = (0,10000) and
   wage_asign_weight_val = (0,1000).

define primitive wa_release with
   wa_release_n = (0,100) and
   wa_release_a = (0,10000) and
   wa_release_weight_val = (0,1000).

define primitive refinanced with
   refinanced_n = (0,100) and
   refinanced_a = (0,10000) and
   refinanced_weight_val = (0,1000).

define primitive cr_cd_lost with
   cr_cd_lost_n = (0,100) and
   cr_cd_lost_a = (0,10000) and
   cr_cd_lost_weight_val = (0,1000).

define primitive clos_inac with
   clos_inac_n = (0,100) and
   clos_inac_a = (0,10000) and
   clos_inac_weight_val = (0,1000).
```

```
define primitive transfered with
   transfered_n = (0,100) and
   transfered_a = (0,10000) and
   transfered_weight_val = (0,1000).

define primitive too_new_rt with
   too_new_rt_n = (0,100) and
   too_new_rt_a = (0,10000) and
   too_new_rt_weight_val = (0,1000).

define primitive paid_satis with
   paid_satis_n = (0,100) and
   paid_satis_a = (0,10000) and
   paid_satis_weight_val = (0,1000).

define primitive paid_acct with
   paid_acct_n = (0,100) and
   paid_acct_a = (0,10000) and
   paid_acct_weight_val = (0,1000).

define primitive cr_ln_clos with
   cr_ln_clos_n = (0,100) and
   cr_ln_clos_a = (0,10000) and
   cr_ln_clos_weight_val = (0,1000).

define primitive deceased with
   deceased_n = (0,100) and
   deceased_a = (0,10000) and
   deceased_weight_val = (0,1000).

define primitive redmd_repo with
   redmd_repo_n = (0,100) and
   redmd_repo_a = (0,10000) and
   redmd_repo_weight_val = (0,1000).

define primitive cur_was_col with
   cur_was_col_n = (0,100) and
   cur_was_col_a = (0,10000) and
   cur_was_col_weight_val = (0,1000).

define primitive cr_ln_rnst with
   cr_ln_rnst_n = (0,100) and
   cr_ln_rnst_a = (0,10000) and
   cr_ln_rnst_weight_val = (0,1000).

define primitive cur_was_for with
   cur_was_for_n = (0,100) and
   cur_was_for_a = (0,10000) and
   cur_was_for_weight_val = (0,1000).
```

```
define primitive pd_not_aa with
   pd_not_aa_n = (0,100) and
   pd_not_aa_a = (0,10000) and
   pd_not_aa_weight_val = (0,1000).

define primitive city_tx_ln with
   city_tx_ln_n = (0,100) and
   city_tx_ln_a = (0,10000) and
   city_tx_ln_weight_val = (0,1000).

define primitive city_tx_rel with
   city_tx_rel_n = (0,100) and
   city_tx_rel_a = (0,10000) and
   city_tx_rel_weight_val = (0,1000).

define primitive consel_ser with
   consel_ser_n = (0,100) and
   consel_ser_a = (0,10000) and
   consel_ser_weight_val = (0,1000).

define primitive co_tax_ln with
   co_tax_ln_n = (0,100) and
   co_tax_ln_a = (0,10000) and
   co_tax_ln_weight_val = (0,1000).



define primitive co_tax_rel with
   co_tax_rel_n = (0,100) and
   co_tax_rel_a = (0,10000) and
   co_tax_rel_weight_val = (0,1000).

/* ------------- END OF CODE ---------------- */
```

APPENDIX B
TRW CREDIT REMARKS

This appendix describes the 102 credit remarks currently used by TRW and employed in MFTES. They were included to broaden the program's documentation and facilitate future maintenance efforts.

ITEMS OF PRIVATE RECORD

INQUIRY:	A copy of the credit profile has been sent to this credit grantor at their request.
CR CD LOST:	Credit card lost or stolen.
CLOSE INAC:	Closed inactive account.
TRANSFERED:	Account transferred to another office.
TOO NEW RT:	Too new to rate.
REFINANCED:	Account renewed or refinanced.
CURR ACCT:	This is either an open or closed account in good standing. If the account is a credit card or charge account, it should be available for use and there may be a balance due. If the account is closed, there were no past due amounts reported and it was paid.
PAID SATIS:	Closed account/paid satisfactory.
PAID ACCT:	Closed account/zero balance/not rated by credit grantor.
CR LN CLOS:	Credit line closed/reason unknown or by consumer request/there may be a balance due.
DECEASED:	Consumer deceased.
CUR WAS DL:	Current account was past due.
CUR WAS 30:	Current account was 30 days past due.
CUR WAS 30-2:	Current account was 30 days past due twice.
CUR WAS 30-3:	Current account was 30 days past due three times.
CUR WAS 30-4:	Current account was 30 days past due four times.
CUR WAS 30-5:	Current account was 30 days past due five times.
CUR WAS 30+6:	Current account was 30 days past due six times or more.

CUR WAS 60:	Current account was 60 days delinquent.
CUR WAS 90:	Current account was 90 days delinquent.
CUR WAS 120:	Current account was 120 days delinquent.
CUR WAS 150:	Current account was 150 days delinquent.
CUR WAS 180:	Current account was 180 days delinquent.
REDMD REPO:	Account was a repossession/now redeemed.
CUR WAS COL:	Current account was a collection account.
CR IN RNST:	Account now available for use and is in good standing. Was a closed account.
CUR WAS FOR:	Current account foreclosure was started.
PD NOT AA:	Paid account. Some payments made past the agreed due dates.
PD WAS 30:	Paid account/was past due 30 days.
PD WAS 30-2:	Paid account/was past due 30 days 2 or 3 times.
PD WAS 30-4:	Paid account/was past due 30 days 4 times.
PD WAS 30-5:	Paid account/was past due 30 days 5 times.
PD WAS 30+6:	Paid account/was past due 30 days 6 times or more.
PD WAS 60:	Paid account/was delinquent 60 days.
PD WAS 90:	Paid account/was delinquent 90 days.
PD WAS 120:	Paid account/was delinquent 120 days.
PD WAS 150:	Paid account/was delinquent 150 days.
PD WAS 180:	Paid account/was delinquent 180 days.
PD COLL AC:	Paid account/was a collection account insurance claim or education claim.
PD REPO:	Paid account/was a repossession.
PD CHG OFF:	Paid account/was a CHARGE-OFF.
PD FORECLO:	Paid account. A foreclosure was started.
PD BY DLER:	Credit grantor paid by company who originally sold the merchandise.
BK LIQ REQ:	Debt included in or discharged through Bankruptcy Chapter 7 or 11.
SETTLED:	Account legally paid in full for less than the full balance.
BK ADJ PLN:	Debt included in or completed through Bankruptcy Chapter 3.
NOT PD AA:	Account not being paid as agreed.
30 DAY DEL:	Account past due 30 days.
30 2 TIMES:	Account past due 30 days 2 times.
30 3 TIMES:	Account past due 30 days 3 times.
30 4 TIMES:	Account past due 30 days 4 times.
30 5 TIMES:	Account past due 30 days 5 times.
30 6+TIMES:	Account past due 30 days 6 times or more.

30 WAS 60:	Account was delinquent 60 days/now 30 days.
DELINQ 60:	Account delinquent 60 days.
DEL WAS 90:	Account was delinquent 90 days/now 60 days.
DELINQ 90:	Account delinquent 90 days.
DEL WAS 120:	Account was delinquent 120 days/now 30, 60 or 90 days.
DELINQ 120:	Account delinquent 120 days.
DELINQ 150:	Account delinquent 150 days.
DELINQ 180:	Account delinquent 180 days.
SCNL NWLOC:	Credit grantor could not locate consumer/consumer now located.
CO NOW PAY:	Now paying, was a charge-off.
FOREPROC:	Foreclosure proceeding started.
GOV CLAIM:	Claim filed with government for insured portion of balance on a loan.
CLOSE NP AA:	Credit line closed/not paying as agreed.
INS CLAIM:	Claim filed for payment of insured portion of balance.
COLL ACCT:	Account seriously past due/account assigned to attorney collection agency or credit grantor's internal collection department.
FORECLOSURE:	Credit grantor sold collateral to settle defaulted mortgage.
VOLUN REPO:	Voluntary repossession.
REPO:	Merchandise was taken back by credit grantor;there may be a balance due.
CHARGE OFF:	Unpaid balance reported as a loss by credit grantor.
SCNL:	Credit grantor cannot locate consumer.

ITEMS OF PUBLIC RECORD

BK 7 FILE:	Voluntary or involuntary Petition in Bankruptcy. Chapter 7-(Liquidation) filed.
BK 7 DISC:	Voluntary or involuntary Petition in Bankruptcy. Chapter 7-(Liquidation) discharged.
BK 7 DISM:	Voluntary or involuntary Petition in Bankruptcy. Chapter 7-(Liquidation) dismissed.
BK 11 FILE:	Voluntary or involuntary Petition in Bankruptcy. Chapter 11-(Reorganization) filed.

BK 11 DISC:	Voluntary or involuntary Petition in Bankruptcy. Chapter 11-(Reorganization) discharged.
BK 11 DISM:	Voluntary or involuntary Petition in Bankruptcy Chapter 11-(Reorganization) dismissed.
BK 13 FILE:	Petition in Bankruptcy Chapter 13-(Adjustment of Debt) filed.
BK 13 DISM:	Petition in Bankruptcy Chapter 13-(Adjustment of Debt) dismissed.
CITY TX LN:	City tax lien
CITY TX REL:	City tax Released
CONSEL SER:	Debt Counseling Service
CO TAX LN:	County Tax Lien
CO TAX REL:	County Tax Released
FED TAX LN:	Federal Tax Lien
FED TX REL:	Federal Tax Released
JUDGMENT:	Judgment
JUDGMT SAT:	Judgment Satisfied
JUDG VACAT:	Judgment Vacated or Reversed
MECH LIEN:	Mechanic's Lien
MECH RELE:	Mechanic's Lien Released
MN MTG FIL:	Manual Mortgage Report (if on written or teleprinter profile, contact your local TRW Information Services Office).
NT RESPON:	Not Responsible Notice, e.q., husband or wife claims not responsible for debts incurred by the spouse.
STAT TX LN:	State Tax Lien
STAT TX REL:	State Tax Released
SUIT:	Suit
SUIT DISMD:	Suit Dismissed or Discontinued
WAGE ASIGN:	Wage Assignment
W/A RELEASD:	Wage Assignment Released

LIST OF REFERENCES

1. Taracad R. Sivasankaran and Tung Bui, "An Expert System to Detect Espionage Activities Through the Analysis of Individual Financial Profiles", Research Paper, Naval Postgraduate School, Monterey, CA., January 1987.

2. Paul M. Rosa, Scrutinize the Finances of Potential Spies, Wall Street Journal, July 23, 1986.

3. Automated Credit Report Assessment - DMDC Matching and Analysis of TRW Credit Reports, prepared by Defense Manpower Data Center, Personnel Security Data and Special Studies Branch, December 1987.

4. Commission to Review DoD Security Policies and Practices, Keeping The Nation's Secrets: A Report To The Secretary of Defense, Department of Defense, Washington, D.C., 1985.

5. Sholom M. Weiss and Casimir A. Kulikowski, A Practical Guide to Designing Expert Systems, Rowman & Allanheld, 1984.

6. Paul Harmon and David King, Expert Systems: Artificial Intelligence in Business, Wiley Press, 1985.

7. Kamal N. Karna, Expert Systems in Government, IEEE Computer Society Press, 1985.

8. Stephen J. Camacho, An Expert System to Assist A Navy Damage Control Assistant With Chemical, Biological, and Radiological Defense, Master's Thesis, Naval Postgraduate School, September 1987.

9. Feliks Kluzniak and Stanislaw Szpakowicz, Prolog For Programmers, p. xi, Academic Press Inc, 1985.

10. Anne Lovett, "Expert Systems Enrich Financial Field", Computerworld, v. 21, pp. 15-17, 2 December 1987.

11. Sivasankaran, T.R., An Expert System Approach To The Analysis Of Individual Financial Profiles, paper presented at WESTEX Conference, Anaheim, CA., June 1987.

12. The Arity/Expert Development Package, 1st ed., v. 1, ARITY CORPORATION, 1986.

13. Kato, M., "14 Product Wrap-Up: Prolog for the People," AI Expert, v. 2, no. 6, pp. 69-69, June 1987.

14. TRW 8022/85A, TRW Glossary of National Status Comments for Verbal Profiles, January 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chief of Naval Operations Director, Information Systems (OP-945) Navy Department Washington, D. C. 20350-2000	2
4.	Department Chairman, Code 52 Department of Computer Sciences Naval Postgraduate School Monterey, California 93943-5000	2
5.	Professor T.R. Sivasankaran, Code 54SJ Naval Postgraduate School Monterey, California 93943-5000	3
6.	Major J. Isett, Code 54IS Naval Postgraduate School Monterey, California 93943-5000	1
7.	LCDR G.M.F. Salazar 7234 W. Pasadena Ave. Glendale, Arizona 85303	2

Thesis
S15265 Salazar
c.1 An expert system to
detect espionage through
credit record analysis.

OCT 91

33512

Thesis
S15265 Salazar
c.1 An expert system to
detect espionage through
credit record analysis.

DEMCO

AMERICAN BINDERY
TOPEKA KANSAS

thesS15265
An expert system to detect espionage thr
3 2768 000 78913 5
DUDLEY KNOX LIBRARY c.1